UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

RECONOCIMIENTO DE EMOCIONES UTILIZANDO LA VOZ EN AMBIENTES DINÁMICOS DE INTERACCIÓN HUMANO-ROBOT

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

NICOLÁS EDUARDO GRÁGEDA USHAK

PROFESOR GUÍA:
NÉSTOR BECERRA YOMA

MIEMBROS DE LA COMISIÓN:
CÉSAR AZURDIA MEZA
ORIETTA NICOLIS

Este trabajo ha sido parcialmente financiado por:
Fondecyt Regular Nº1211946

SANTIAGO DE CHILE
2023

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIAS
DE LA INGENIERIA, MENCIÓN ELÉCTRICA
Y MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO
POR: NICOLÁS EDUARDO GRÁGEDA USHAK
FECHA: 2023
PROF. GUÍA: NÉSTOR BECERRA YOMA

RECONOCIMIENTO DE EMOCIONES UTILIZANDO LA VOZ EN AMBIENTES DINÁMICOS DE INTERACCIÓN HUMANO-ROBOT

La capacidad de los robots de reconocer emociones de sus usuarios es esencial para que exista una interacción humano-robot natural. En este trabajo de tesis se estudia el reconocimiento automático de emociones utilizando la voz en ambientes reales de interacción humano-robot. Los problemas derivados de esta interacción real afrontados en esta tesis son: el efecto del canal acústico a la señal de voz, la presencia de ruido ambiental, el ruido mecánico interno del robot y la existencia de movimiento relativo entre el usuario y el robot, lo que genera que el canal acústico sea dinámico. Se propone un sistema que utiliza información de la posición del usuario para aplicar *beamforming*, en conjunto a modelos *deep learning* con el mejor rendimiento en reconocimiento de emociones del estado del arte. El sistema mencionado logra una mejora de un 30 % en términos de *Concordance Correlation Coefficient* (CCC) para la prueba de interacción humano-robot (HRI) estática al compararse con el caso base (utilizando sólo el mejor modelo de reconocimiento de emociones entrenado con la base de datos original). Por otro lado, con el mismo sistema se obtiene una mejora de un 24 % para la prueba dinámica de HRI al compararse con el caso base.

En la Ciencia la única verdad sagrada,
es que no hay verdades sagradas.

Carl Sagan

Agradecimientos

Estoy sumamente agradecido con mis padres, todo este trabajo no podría ser posible sin su apoyo incondicional, no solo durante mi etapa universitaria, sino a lo largo de toda mi vida. A mi hermano Andrés, mis abuelos y a toda mi familia que siempre me han alentado a conseguir mis metas.

Agradezco a Javiera Cáceres, un pilar fundamental para mi durante esta etapa, gracias por el apoyo y por todos los buenos momentos, siempre estaré agradecido por haberte conocido.

También, agradezco a todos los amigos que hice durante mi paso por la Universidad. En plan común me acogieron y me hicieron sentir como que hubiese toda mi vida en Santiago, luego durante la especialidad tuve la oportunidad de conocer aún más personas geniales, con las que pasamos juntos penas y alegrías durante tantas noches de estudio (y fiestas). Agradezco a todos mis compañeros del Laboratorio de Procesamiento y Transmisión de Voz (LPTV), que me acompañaron e hicieron ameno el tramo final de mi etapa universitaria. Al profesor Néstor Becerra Yoma, que me guió en mis primeros pasos en el campo de la investigación.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivación

En los próximos 10 a 20 años, se espera que la colaboración entre humanos y robots sea una parte importante de las aplicaciones empresariales. Por lo tanto, la robótica social es uno de los desafíos más críticos en la ciencia y la ingeniería robótica. Aunque se han hecho algunos avances en este campo, todavía no es posible la interacción social completa entre humanos y robots en condiciones reales. Uno de los escenarios más difíciles y poco explorados ocurre cuando los usuarios, que pueden estar en movimiento, intentan interactuar con un robot también en movimiento. Además, esta interacción puede ocurrir en ambientes ruidosos, lo que dificulta la comunicación.

La interacción fluida con los humanos es de gran dificultad para los robots, ya que requiere que reconozcan expresiones faciales, contenido lingüístico y prosodia hablada para comportarse de manera adecuada. La comunicación a través del lenguaje es una herramienta importante ya que transmite una gran cantidad de información lingüística y paralingüística sobre los estados psicológicos, físicos y emocionales de los seres humanos. A pesar de lo anterior, en los últimos años, gracias a los avances en inteligencia artificial, han existido intentos de entregarle a los robots la capacidad de expresar y detectar emociones de forma artificial.

En particular, el área del reconocimiento de emociones utilizando la voz o *Speech Emotion Recognition* (SER) es de alta relevancia debido a que permite a los ordenadores entender y responder a las emociones humanas de una forma más natural. Esto puede tener una amplia gama de aplicaciones, como la interacción persona-ordenador, los asistentes virtuales, la atención al cliente, entre otras. Por ejemplo, en el servicio de atención al cliente, un sistema informático capaz de reconocer el estado emocional de un cliente, como la frustración o la ira, puede responder de forma más adecuada y empática, mejorando así la experiencia del cliente.

Sin embargo, a pesar de que existe consenso sobre la importancia del *Human Robot Interaction* (HRI) móvil en los robots sociales, no existen estudios que analicen el efecto de este movimiento sobre el canal acústico en sistemas de reconocimiento de emociones que utilicen la voz como entrada principal.

1.2. Hipótesis de trabajo

En esta investigación se define la siguiente hipótesis:

- Es posible reducir la disparidad de rendimiento entre las condiciones de HRI y aquellas originales con degradación acústica controlada de un sistema de reconocimiento de emociones mediante el entrenamiento en condiciones acústicas similares a las de prueba y el uso de técnicas como *beamforming*.

1.3. Objetivo general

El objetivo general de este trabajo es el de mejorar el rendimiento de sistemas de reconocimiento de emociones por voz en ambientes reales y complejos de interacción humano-robot con respecto al sistema *baseline* sin el uso de técnicas de *beamforming* ni modelamiento del canal acústico.

1.4. Objetivos específicos

Para cumplir con el objetivo general se definen los siguientes objetivos específicos:

- Grabar una base de datos de evaluación para reconocimiento de emociones en ambientes de HRI reales. Esta quedará como un aporte para la posteridad.
- Evaluar y reducir el deterioro de técnicas de reconocimiento de emociones con voz del estado del arte en ambientes de HRI estáticos.
- Evaluar y reducir el deterioro de técnicas de reconocimiento de emociones con voz del estado del arte en ambientes de Recorded dinámicos.

Capítulo 2

Marco teórico y antecedentes bibliográficos

En esta capítulo se definirán los fundamentos teóricos, así como conceptos esenciales para comprender el trabajo de tesis. Este apartado se divide en tres secciones:

- Robots sociales y perfilamiento de usuario en HRI, donde se detalla las razones y estudios que fundamentan la importancia de la capacidad de un robot para crear de un perfil psicológico por usuario.
- Reconocimiento automático de emociones con voz, donde se muestra los antecedentes bibliográficos y conceptos utilizados para la elaboración este trabajo, asociados al reconocimiento automático de emociones.
- Herramientas y aplicaciones que utilizan la voz en HRI, muestra la revisión bibliográfica y herramientas de procesamiento de señales de voz usados en esta tesis.

2.1. Robots sociales y perfilamiento de usuario en HRI

La colaboración sin fisuras entre humanos y robots será un componente estratégico en las aplicaciones comerciales de los próximos 10 a 20 años. En consecuencia, la robótica social es uno de los retos más importantes y críticos de la ciencia y la ingeniería robótica. Aunque se ha avanzado algo en este tema, hoy en día no es posible una interacción social completa entre humanos y robots en condiciones reales. Uno de los escenarios más desafiantes y menos explorados es aquel en el que uno o más usuarios, que pueden estar en movimiento, intentan interactuar con el robot, que también puede estar en movimiento. Además, esta interacción puede producirse en un entorno ruidoso, lo que afecta a esta comunicación.

La interacción social es un reto muy complejo para la robótica, en parte porque requiere reconocer o detectar eficazmente las direcciones de la mirada, las expresiones faciales y el contenido lingüístico y la prosodia del habla, para luego actuar en consecuencia. Dependiendo del contexto cultural, la diferencia entre los estados emocionales humanos puede ser tan sutil como «un simple guiño, o una inflexión ascendente en un solo fonema» [1]. Para lograr este propósito, los sistemas tendrán que combinar múltiples modalidades de entrada. Sin embargo, algunas de estas entradas, como las señales fisiológicas, requieren sensores portátiles que pueden resultar invasivos desde el punto de vista del usuario. Además, el procesamiento de imágenes no siempre es posible en función de las condiciones de funcionamiento. En

cambio, el habla transmite una enorme cantidad de información lingüística y paralingüística (por ejemplo, prosodia). Más allá de las órdenes de voz a los robots, el habla es una ventana al estado psicológico, físico y emocional de los seres humanos.

La elaboración de perfiles de usuarios emocionales es esencial para HRI, ya que se espera que los robots sean capaces de reconocer las intenciones y los objetivos que subyacen a las acciones del usuario, con el fin de adaptar su comportamiento a ellos [2]. Además, el perfil social también se refiere a la capacidad de reconocer fenómenos sociales, como el compromiso, el conflicto, la empatía, el interés y las emociones, que no pueden observarse directamente, sino que deben inferirse examinando indicadores indirectos. Algunos de estos indicadores indirectos pueden ser la postura corporal [3], las expresiones faciales [4], [5], la dirección de la mirada [6], [7], el volumen de la voz, etcétera. Dentro del *emotion user profiling*, surge el concepto de *emotion recognition*, que busca detectar dinámicamente el estado emocional del usuario durante la interacción, ya que, mientras que el perfil emocional de una persona no cambia durante una única interacción con el robot, el usuario puede mostrar múltiples emociones durante la interacción. Esta detección continua permite actualizar el perfil del usuario.

2.2. Reconocimiento automático de emociones con voz

2.2.1. Antecedentes del reconocimiento automático de emociones utilizando la voz

En "*Survey of Emotions in Human-Robot Interactions: Perspectives from Robotic Psychology on 20 Years of Research*" [8] se concluye que el estudio de las emociones en HRI es cada vez más importante a medida que los robots sociales se hacen más frecuentes en nuestra vida cotidiana. Además, el autor de esta revisión plantea que si se desarrollan sistemas capaces de reconocer las emociones con precisión, se podrán crear sistemas de HRI más eficaces, capaces de adaptarse al estado emocional del usuario, incluso, estos podrán usarse en aplicaciones en que ayuden a comprender mejor cómo perciben y expresan las emociones los seres humanos, como la psicología y la neurociencia.

En la misma línea, el artículo "*Emotion Recognition for Human-Robot Interaction: Recent Advances and Future Perspectives*" [9] concluye que el reconocimiento de emociones es un aspecto importante de la interacción humano-robot y que puede mejorar la comunicación entre estos. Los autores plantean que, a pesar de que los recientes avances en el aprendizaje automático y la visión por ordenador han propiciado progresos significativos en el reconocimiento de emociones, aún quedan retos por abordar tales como la mejora de la precisión, la robustez y la adaptabilidad a distintos contextos. También, se plantea el reto del desarrollo de sistemas multimodales de reconocimiento de emociones más eficaces, capaces de combinar distintas fuentes de información, como las expresiones faciales, el habla y las señales fisiológicas.

Con respecto al reconocimiento de emociones multimodal, los autores de "*A systematic survey on multimodal emotion recognition using learning algorithms*" [10] identifican varias lagunas de investigación y oportunidades para futuras investigaciones. Entre estas lagunas se encuentra la necesidad de disponer de conjuntos de datos más estandarizados, la falta

de investigación sobre las diferencias transculturales en el reconocimiento de emociones y la necesidad de más investigación sobre el reconocimiento de emociones en tiempo real. Por otro lado, algunas de las oportunidades identificadas por los autores incluyen la investigación de modalidades para el reconocimiento de emociones menos exploradas, como las señales fisiológicas o la prosodia de la voz, y el desarrollo de nuevas técnicas de fusión para combinar múltiples modalidades.

El proceso de identificar emociones humanas utilizando la voz, principalmente elementos no verbales de la voz, se define como reconocimiento de emociones por voz o *Speech Emotion Recognition* (SER). La importancia del desarrollo de SER en HRI es evidente; en "*The Necessity of Emotion Recognition from Speech Signals for Natural and Effective Human-Robot Interaction in Society 5.0*" [11] se nombra dentro de las importancias la mejora en la capacidad de los robots para interactuar con los humanos de forma más natural y eficaz. Esto se debe a que las emociones desempeñan un papel importante en la comunicación humana, y ser capaz de reconocer y responder a las emociones puede ayudar a los robots a entender y responder mejor a las necesidades humanas. Además, se menciona que SER en HRI es un campo que, a pesar de que prácticamente todos los robots sociales tienen micrófonos, sorprendentemente casi no ha sido explorado en la literatura.

La gran mayoría de las investigaciones en esta disciplina (reconocimiento de emociones con voz) se centran en la interacción humano-computador o *Human-Computer Interaction* (HCI) [12], por lo que pueden despreciar el efecto del canal acústico gracias a que en este tipo de aplicación el usuario está al lado del micrófono. Más aún, existen revisiones bibliográficas enteras sobre técnicas de SER que no mencionan la aplicación de estas en HRI [13].

En [12] se muestra que sólo unos pocos estudios han probado la tarea de SER a distancia y en entornos ruidosos. Las técnicas más utilizadas para abordar este reto son la selección de características que sean más robustas a las distorsiones por distancia y la creación de modelos codificador-decodificador, que se sabe que son robustos en tareas que implican varios tipos de distorsiones.

Los autores de "*Distant emotion recognition*" [14] consiguen seleccionar 48 descriptores de bajo nivel (LLD), que se extraen por fotograma y se pasan por una *Long Short Term Memory* (LSTM) para la clasificación final. El entorno de prueba de este estudio es una sala de reuniones con siete micrófonos fijos distribuidos por la sala; se realiza filtrado espectral y temporal. Sin embargo, no se utiliza ninguna técnica de *beamforming*.

Por su parte, los autores de "*Real time distant speech emotion recognition in indoor environments*" [15] utilizan una métrica para determinar la distorsión de las características en función de la distancia al micrófono. Además, entrenan su clasificador con audio convolucionado con *Room Impulse Responses* (RIRs) generados artificialmente y utilizan el algoritmo WPE para eliminar la reverberación de los audios de prueba y *Coherent-to-Diffuse Power Ratio Estimation* (CDR) para realizar el *de-noise*. Sin embargo, en este estudio sólo se evalúan situaciones estáticas, variando la distancia al micrófono.

En "*Two-layer fuzzy multiple random forest for speech emotion recognition in human-robot interaction*" [16] se evalúa una técnica de adquisición de rasgos utilizando una plataforma

robótica con un Kinect montado. Sin embargo, la base de datos de prueba está actuada por voluntarios de su propio laboratorio de investigación y sólo tiene 500 enunciados. Además, los autores no utilizan ninguna técnica de mejora del habla, ni evalúan el efecto del movimiento del robot.

2.2.2. Reconocimiento dimensional de emociones

El reconocimiento categórico de las emociones es el proceso de categorizar el habla en un número restringido de categorías predeterminadas, como feliz, triste, enojado, etcétera. Esta estrategia es básica y fácil de adoptar, y puede producir resultados claros y comprensibles.

Sin embargo, la complejidad de las emociones humanas no puede ser descritas mediante el uso de unas pocas etiquetas categóricas [17]. Es por esto que, en lugar de clasificar el habla en categorías predeterminadas, el reconocimiento dimensional de emociones incluye la estimación de los valores continuos de atributos emocionales como la valencia o *valence* (emoción positiva o negativa) y la excitación o *arousal* (grado de excitación).

En esta tesis se utiliza el reconocimiento de emociones en un espacio de tres dimensiones continuas de atributos emocionales (*arousal*, *dominance* y *valence*). Este método puede producir respuestas más sutiles y exhaustivas, pero es más difícil de aplicar y puede requerir más datos y recursos de procesamiento.

2.2.2.1. *Concordance Correlation Coefficient* (CCC)

Debido a que en el reconocimiento de emociones dimensional no se puede usar métricas categóricas como la precisión o *accuracy*, se debe buscar una métrica que mejor se ajuste para medir el rendimiento del sistema. En esta línea, el *Concordance Correlation Coefficient* (CCC) es un coeficiente que mide de la concordancia entre dos conjuntos de mediciones continuas [18]. Tiene una escala de -1 a 1, donde cerca de 1 significa un fuerte acuerdo y cerca de -1 sugiere un grave desacuerdo. El CCC posee numerosas características clave: es simétrico, lo que significa que el orden de las mediciones no afecta al valor del CCC; también es invariante respecto a la escala, lo que significa que los cambios en las unidades de medida no afectan al CCC; y además, el CCC es inmune a los valores atípicos, lo que lo convierte en una medida robusta de la concordancia.

La ecuación que calcula el CCC de un conjunto de pares ordenados (x_n, y_n) de largo N se muestra a continuación:

$$CCC = \frac{2s_{xy}}{s_x^2 + s_y^2 + (\bar{x} - \bar{y})^2} \tag{2.1}$$

Donde $\bar{x}$ y $\bar{y}$ son los promedios de x e y, respectivamente. Además. s_x^2 y s_y^2 corresponden a varianzas y s_{xy} es la covarianza calculada:

$$s_{xy} = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(y_n - \bar{y}) \tag{2.2}$$

2.2.3. Inteligencia artificial: conceptos y modelos asociados a SER

2.2.3.1. Red neuronal

Las redes neuronales artificiales, o simplemente redes neuronales, son un conjunto de capas de nodos de procesamiento con conexiones entre si. La inspiración de estos modelos es el cerebro humano, ya que este consiste en neuronas conectadas a través de axones que permiten la sinapsis.

2.2.3.2. Perceptrón multicapa

Un perceptrón (ver Figura 2.1) es la unidad de procesamiento básica de un modelo de aprendizaje automático. Se puede utilizar de forma individual para clasificar de forma binaria. Recibe el vector de características del dato de entrada, para luego definir la salida de este en función de si la suma de las entradas multiplicada por sus pesos asociados alcanza un determinado umbral. Las ponderaciones y el umbral se aprenden a partir de los datos de entrenamiento modificando las ponderaciones para minimizar la diferencia entre la salida esperada y la real.

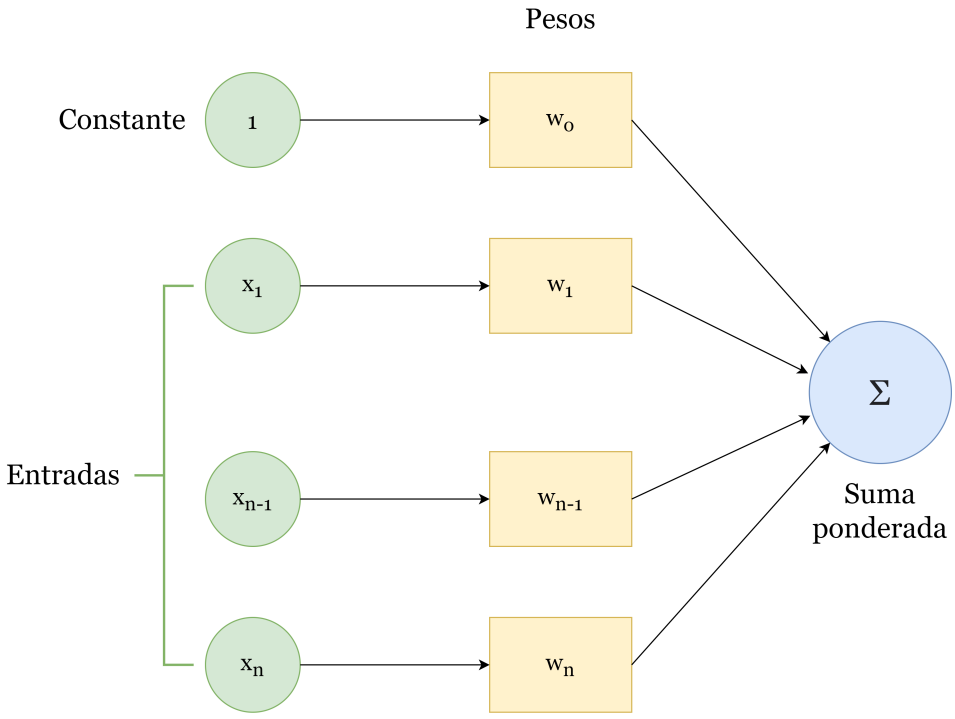


Figura 2.1: Diagrama básico de un perceptrón.

El perceptrón multicapa o *multilayer perceptron* (MLP) es un tipo de red neuronal artificial. Suele constar de varias capas de ”neuronas” artificiales, que se conectan y activan de una forma específica para procesar y analizar los datos de entrada. Cada una de estas neuronas es un perceptrón, al cual se le aplica alguna función no-lineal a su salida para limitar el rango de su recorrido. EL MLP consta de una primera capa, llamada capa de entrada, una o más capas intermedias, llamadas capas ocultas, y una capa final, llamada capa de salida (ver Figura 2.2). Debido a que los perceptrones de cada capa entregan su salida a todas las neuronas de la siguiente capa, este tipo de capas se llaman también *Fully Connected* (FC).

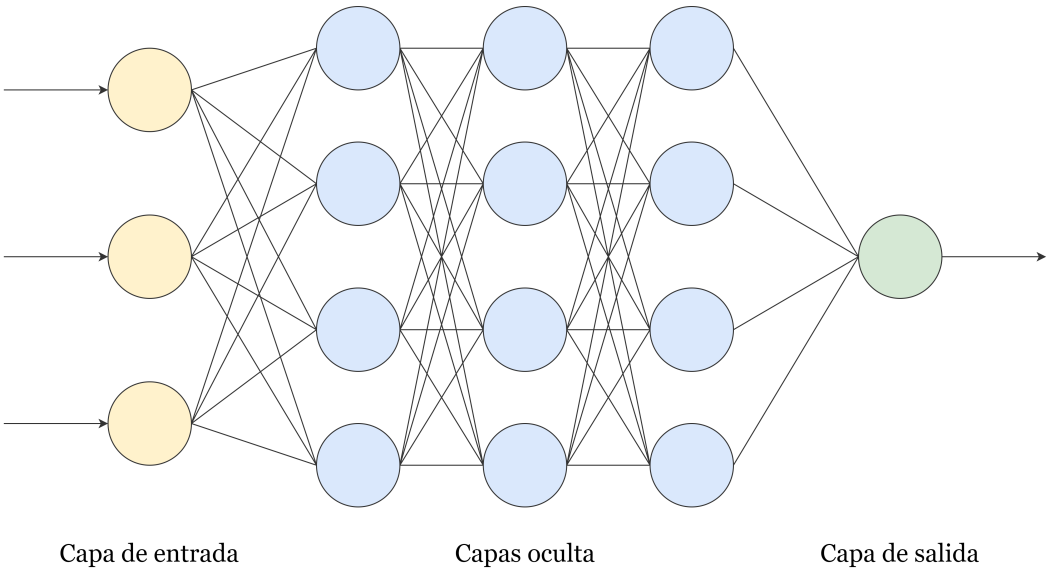


Figura 2.2: Diagrama de un MLP de tres capas ocultas.

2.2.3.3. Red neuronal convolucional

Las redes neuronales convolucionales, o *Convolutional Neural Networks* (CNN) en inglés, son un tipo de red neuronal artificial que utiliza la convolución en lugar de la multiplicación matricial tradicional. Se crearon expresamente para analizar arreglos de datos, ya sean vectores 1D (series de tiempo), matrices 2D (como imágenes) o tensores 3D (como videos).

Capas convolucionales

Las capas convolucionales son un componente fundamental de las CNN. Una capa convolucional aplica un conjunto de filtros a la entrada, donde cada filtro es una pequeña matriz entrenada para detectar una característica específica en la entrada, como bordes, texturas o formas. Los filtros se convolucionan con la entrada, es decir, se selecciona una porción de la entrada del mismo tamaño que el filtro y se multiplica término por término con este, para luego sumar todos estos resultados. Este proceso se repite para cada posición de la entrada y para cada filtro, a fin de producir múltiples mapas de características. Las dimensiones de las entradas y los filtros pueden variar, en las Figuras 2.3 y 2.4 muestran ejemplos de como funciona la aplicación de filtros en una y dos dimensiones, respectivamente.

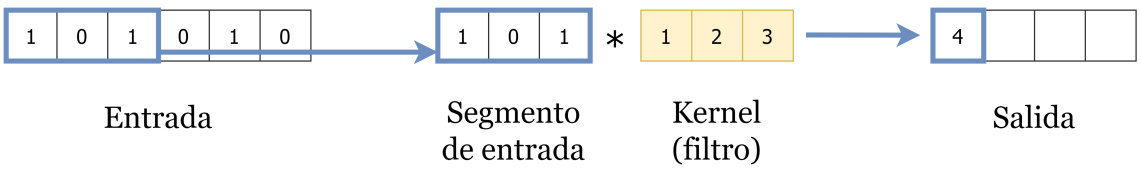


Figura 2.3: Funcionamiento de convolución en una dimensión.

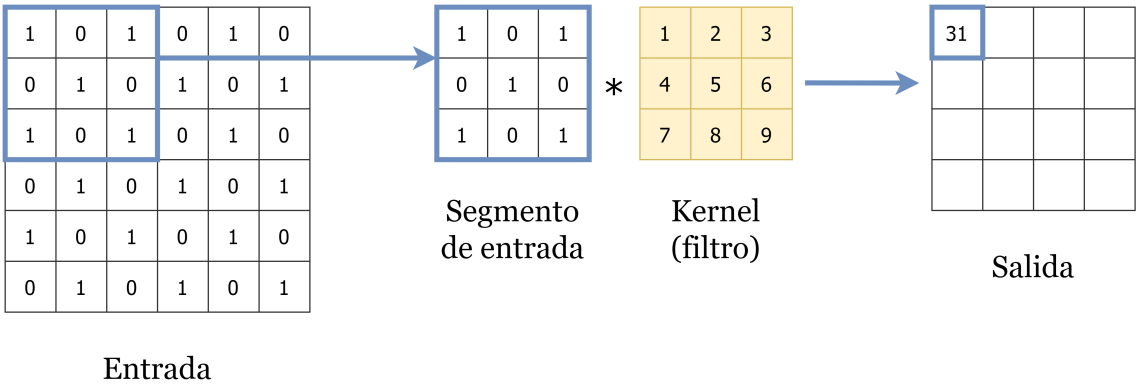


Figura 2.4: Funcionamiento de convolución en dos dimensiones.

Una de las principales ventajas de las capas convolucionales es su capacidad para aprender patrones locales en la entrada, una propiedad útil para tareas en las que la entrada es un arreglo (vector, matriz o tensor) con cierto nivel de correlación entre sus elementos aledaños. Además, los pesos compartidos de los filtros en la entrada permiten a la red aprender de forma más eficiente y generalizar mejor.

En las CNN, se apilan varias capas convolucionales, cada una con un conjunto diferente de filtros y parámetros, para extraer características de alto nivel de la entrada.

El cálculo del largo de la salida de una capa *Conv1D* se describe a continuación. Sea una entrada con dimensiones (C_{in}, L_{in}) y la salida de una capa de convolución 1D con salida (C_{out}, L_{out}), entonces:

$$L_{out} = \frac{L_{in} + 2 \times \ padding \ - \ dilation \ \times (kernel - 1) - 1}{stride} + 1 \tag{2.3}$$

Capas de agrupación

Las CNNs también incluyen una operación de agrupación o *pooling*, que se utiliza para reducir las dimensiones espaciales del mapa de características, conservando la información más importante. La operación de agrupación más común es el *max pooling*, que consiste en seleccionar el valor máximo de una pequeña porción del mapa de características. El *pooling* ayuda a reducir el coste computacional y a aumentar la robustez de la red.

2.2.3.4. Autoencoder

Un *autoencoder* es un tipo de red neuronal que se utiliza para aprender una representación (codificación) de un conjunto de datos, normalmente con el fin de reducir la dimensionalidad o reducir el ruido presente. Un *autoencoder* consta de dos partes: un codificador, que mapea los datos de entrada a una representación de menor dimensión, y un decodificador, que mapea la representación de menor dimensión de vuelta al espacio original. El objetivo del *autoencoder* es aprender una representación (codificación) que capture las características esenciales de los datos, reduciendo al mismo tiempo la dimensionalidad. El autoencoder puede estar formado por distintos tipos de capas, ya sean capas de MLP, convolucionales, recurrentes o atencionales.

2.2.3.5. Ladder network

Una *ladder network* es un tipo de modelo de *deep learning* que se introdujo por primera vez por Rasmus et al. en el trabajo "*Semi-supervised learning with ladder networks*" [19]. La arquitectura de una *ladder network* combina las fortalezas del aprendizaje supervisado y no supervisado mediante su estructura de eliminación de ruido tipo *autoencoder*.

Una *ladder network* consta de dos componentes principales: el *autoencoder denoising*, entrenado de forma no supervisada, y el MLP, que se entrena de forma supervisada. A su vez, el *denoising autoencoder* se compone de un codificador y un decodificador con conexiones laterales entre ellos.

Durante el entrenamiento, los datos de entrada se corrompen agregando ruido blanco gaussiano y el *denoising autoencoder* se entrena para reconstruir los datos originales a partir de la versión corrompida. Esto obliga al *denoising autoencoder* a aprender a extraer una representación robusta de los datos de entrada, es decir, que sea capaz de tolerar el ruido.

El MLP se entrena al mismo tiempo que el *autoencoder* para resolver la tarea principal (normalmente esta tarea es de clasificación o de regresión). Utiliza como entrada la salida del *encoder* del *denoising autoencoder*, permitiendo que la red supervisada se beneficie de la representación robusta de los datos de entrada aprendida por el *denoising autoencoder*.

La arquitectura de *ladder network* es especialmente útil para tareas de aprendizaje semi-supervisado, en las que sólo se dispone de una pequeña cantidad de datos etiquetados. El *autoencoder* de eliminación de ruido no supervisado es capaz de aprender una representación útil de los datos de entrada a partir de la gran cantidad de datos sin etiquetar, que luego puede utilizarse para mejorar el rendimiento del MLP en los datos etiquetados.

En conclusión, las *ladder network* son un tipo de modelo de *deep learning* potente y versátil que utiliza aprendizaje supervisado y no supervisado, lo cual ha provocado que sea una red sumamente utilizada en el reconocimiento de emociones [20][21][22][23]. En concreto, los autores de "*Separation of emotional and reconstruction embeddings on ladder network to improve speech emotion recognition robustness in noisy conditions*" [24] prueban una implementación de *ladder network* en un entorno ruidoso, utilizando el micrófono de un smartphone, un altavoz que reproduce el habla y otro en un extremo opuesto que reproduce el ruido, demostrando su robustez en escenarios de uso real.

2.2.3.6. Redes atencionales

La atención en las redes neuronales artificiales está diseñada para reproducir el modo en que el cerebro humano presta atención. Enfatiza ciertas partes de la información y resta importancia a otras, con el objetivo de dirigir más la atención a los aspectos significativos, aunque sutiles, de los datos. La determinación de qué partes de los datos son más significativas que otras depende del contexto. Es importante tener en cuenta que existen varios tipos de mecanismos de atención y que cada uno puede visualizarse de forma diferente. Pero, por lo general, los mecanismos de atención se utilizan para centrarse en elementos específicos de la entrada en lugar de procesar toda la entrada por igual.

La atención es un método utilizada en las redes neuronales para centrarse de forma se-

lectiva en fragmentos específicos de información mientras se procesa. Suele utilizarse en aplicaciones de procesamiento del lenguaje natural (PLN), como la traducción automática, el resumen de textos y la respuesta a preguntas. El mecanismo de atención permite al modelo priorizar distintas secciones de la entrada, lo que le permite centrarse en la información más importante mientras procesa la entrada.

La noción principal de la atención es asignar un peso a cada componente de la entrada, reflejando su relevancia en el resultado final. Estos pesos se utilizan para calcular una suma ponderada de la entrada, que posteriormente se utiliza para generar la salida final. El mecanismo de atención más simple puede expresarse matemáticamente del siguiente modo:

$$y = \sum_{i=1}^{n} \alpha_i xi \tag{2.4}$$

Donde y es la salida final, x_i es el i-ésimo elemento de la entrada, y α_i es el peso asignado a ese elemento.

Existen varios tipos de mecanismos de atención, pero uno de los más comunes es la atención de producto punto. El peso asignado a cada componente de la entrada en esta forma de atención se calcula como el producto punto de un vector de *query* (Q), un vector *key* (K) y un vector de valor (V). El vector *query* representa la información en la que el modelo desea centrarse, el vector *key* refleja la información de la entrada y el vector valor indica la importancia de la información de la entrada. La fórmula de esta operación se muestra en la ecuación 2.5.

$$\text{Atencion}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.5}$$

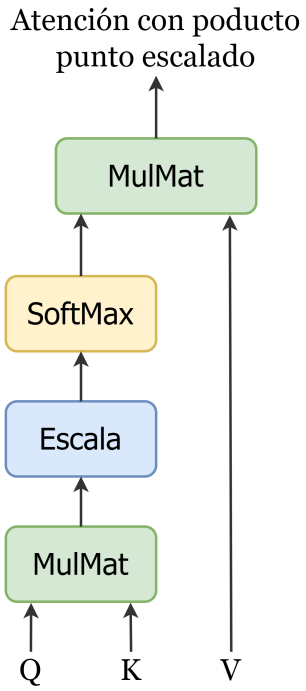


Figura 2.5: Mecanismo de atención de producto punto.

Otro tipo de atención es la atención multicabezal, que permite al modelo atender a varias secciones de la entrada al mismo tiempo. En esta forma de atención, el modelo emplea varios vectores de *query*, *key* y valor, calculando los pesos de atención para cada conjunto de vectores individualmente. El resultado final se calcula como una concatenación de los resultados de cada cabezal de atención.

$$\text{Multicabeza}(Q, K, V) = \text{Concat}\left(\text{cabeza}_1, \ldots, \text{cabeza}_\text{h}\right) W^O \tag{2.6}$$

Donde:

$$\text{cabeza}_i = \text{Atencion}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2.7}$$

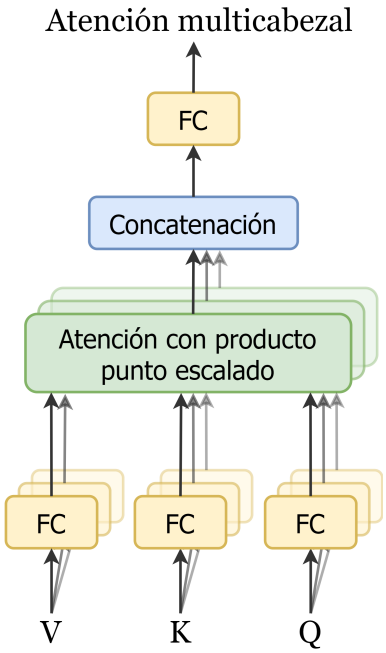


Figura 2.6: Mecanismo de atención multicabezal.

En general, la atención es un poderoso mecanismo que permite a las redes neuronales centrarse selectivamente en determinadas partes de la entrada mientras la procesan. En los últimos años, esta herramienta ha cobrado mayor relevancia gracias a su uso en las redes neuronales tipo *transformer*.

2.2.3.7. Redes neuronales tipo *transformer*

Una red neuronal tipo *transformer* es un tipo de modelo de *deep learning* propuesto en [25] que ha ganado una popularidad significativa en los últimos años debido a su capacidad para manejar datos secuenciales, como el lenguaje natural y los datos de series temporales.

Una red neuronal transformadora es, en esencia, un diseño codificador-decodificador. El codificador procesa la secuencia de entrada a través de una serie de capas, cada una de las cuales tienen un mecanismo de autoatención y una red neuronal, entregando un conjunto de estados ocultos, los cuales luego se transfieren al decodificador produciendo así la secuencia de salida.

Al generar estados ocultos, la autoatención permite al modelo sopesar la relevancia de las distintas secciones de la secuencia de entrada. Esto se consigue calculando un conjunto de pesos de atención para cada lugar de la secuencia de entrada, ponderando la contribución de cada posición a los estados ocultos, logrando identificar aquellas secciones más relevantes para la salida.

Además, el modelo es capaz de prestar atención a distintas partes de la secuencia de entrada en paralelo, esto gracias al mecanismo de atención multicabezal. Permitiéndole al modelo aprender diferentes representaciones de esta. Esto es útil para tareas como la comprensión del lenguaje.

Gracias a todo lo anterior, la arquitectura del transformador se ha utilizado en una amplia gama de tareas de procesamiento del lenguaje natural, como la traducción de idiomas, el resumen de textos y la respuesta a preguntas. También ha sido utilizado en otros ámbitos del reconocimiento del habla y la predicción de series temporales.

La arquitectura de los *transformers* también se ha utilizado en modelos preentrenados como BERT, GPT-2 y GPT-3, que han demostrado ser muy eficaces en una amplia gama de tareas de procesamiento de lenguaje natural. Estos modelos se preentrenan con grandes cantidades de datos y luego se ajustan a tareas específicas. Se ha demostrado su eficacia al momento de transferir los conocimientos adquiridos durante el preentrenamiento a la tarea final.

En conclusión, las redes neuronales *transformers* son un tipo de modelo potente y versátil que ha demostrado ser eficaz en una amplia gama de tareas de datos secuenciales. El uso de mecanismos de autoatención y atención multicabezal permite al modelo manejar secuencias de longitudes variables y aprender diferentes representaciones de los datos de entrada. Esto hace que la arquitectura transformadora sea una opción popular en los últimos años para el reconocimiento de emociones utilizando la voz.

2.2.3.8. Wav2vec 2.0

Wav2vec 2.0 [26] es un tipo de arquitectura de *deep learning* que es capaz de procesar audios de forma directa, es decir, sin ningún tipo de extracción de características inicial. En el último periodo ha logrado superar el estado del arte para tareas como *speech to text* y reconocimiento de emociones utilizando la voz.

El modelo consta de dos partes: un codificador convolucional multicapa y una red tipo *transformer*. El codificador toma la señal de audio sin procesar y genera representaciones latentes del habla para múltiples pasos temporales. Estas representaciones se introducen en la red tipo *transformer* para crear representaciones contextuales de toda la secuencia. La salida del codificador de características también se cuantifica en un conjunto finito de representaciones del habla para el entrenamiento autosupervisado. Tanto el codificador como el transformador utilizan diversas técnicas, como la normalización, las funciones de activación y las capas convolucionales, para mejorar el rendimiento. Además, el *transformer* utiliza una capa convolucional como incrustación posicional relativa en lugar de incrustaciones posicionales fijas.

En la Figura 2.7 se muestra el diagrama de una arquitectura Wav2vec 2.0.

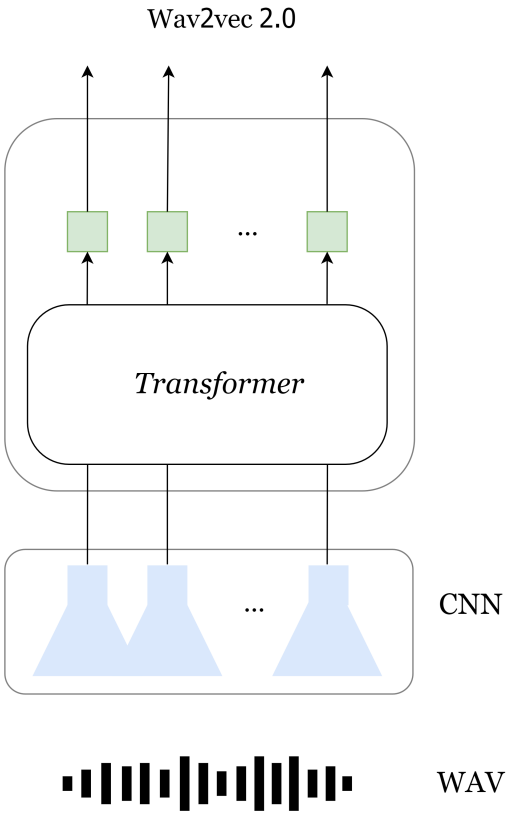


Figura 2.7: Diagrama generalizado de Wav2vec 2.0.

Codificador de características con CNN

El codificador de características se compone de tres bloques: el primero consiste en una capa convolucional 1D, el que sigue en una capa de normalización, y el último, en una función de activación tipo *Gaussian Error Linear Unit* (GELU). El archivo de audio que entra al codificador se normaliza previamente a media cero y varianza unitaria.

Bloque de *transformer*

La salida del codificador de características CNN es procesada posteriormente por una red tipo *transformer* con una arquitectura similar a la propuesta en "*Transformers with convolutional context for asr*" [27]. En lugar de representaciones de posición fijas, las cuales codifican la información posicional de manera absoluta, Wav2vec 2.0 utiliza una capa convolucional que funciona como una representación posicional relativa.

2.3. Herramientas y aplicaciones que utilizan la voz en HRI

2.3.1. Antecedentes de reconocimiento de emociones en HRI

Los robots móviles son cruciales en HRI, tanto en tareas industriales [28][29] como en tareas de mayordomo o asistente personal [16][30]. Increíblemente, hasta ahora no ha sido probado el rendimiento de modelos de SER en escenarios móviles de HRI. Sin embargo, sí

existen estudios sobre el efecto del dinamismo en HRI para la tarea de "*speech to text*" [31]. Basándose en estos estudios, se propone una configuración similar para re-grabar la partición de test de nuestra base de datos. La configuración del ambiente de prueba propuesto ilustra el problema genérico de la HRI en la robótica móvil puesto que presenta los problemas de reconocimiento distante en un ambiente con dos fuentes externas de ruido y el ruido proveniente de los motores del robot. Además, aprovechando los sensores montados en el robot, se usan técnicas de filtrado espacial o *beamforming* que utilizan matrices de micrófonos para dirigir el lóbulo principal de esta matriz hacia la fuente de habla objetivo o el usuario.

2.3.2. *Beamforming*

Beamforming es un conjunto de técnicas de filtrado espacial que se utilizan para mejorar las señales que provienen de una determinada dirección en relación con un conjunto de dos o más micrófonos, reduciendo el ruido y la interferencia provenientes de otras direcciones. Sin embargo, la capacidad de los enfoques tradicionales de *beamforming* para disminuir la reverberación y el ruido difuso es limitada [32]. Este fenómeno es principalmente el resultado de la dificultad de estimar con precisión los retrasos en entornos reverberantes. Como resultado, las técnicas de *beamforming* son menos aplicables en ambientes interiores donde muchos reflejos de las paredes pueden producir un campo difuso o reverberante. Sin embargo, en los trabajos de Novoa et al. [31] y Díaz et al. [33] se utilizan y comparan el uso de distintas técnicas de *beamforming* para un sistema de *Automatic Speech Recognition* (ASR) en una plataforma robótica, logrando mejoras con respecto a los casos base.

2.3.2.1. *Direction of Arrival* (DOA)

El *Direction of Arrival* (DOA) se define como la dirección de la que proviene una determinada onda con respecto a un sensor. En el caso de este estudio, se trata del ángulo que forma la dirección de propagación de la onda de sonido con el arreglo de micrófonos.

2.3.2.2. *Delay and Sum*

Un método de beamforming muy conocido es el *Delay and Sum* (D&S) [34]. El enfoque de este consiste en sumar las señales retrasadas para dirigir la dirección de ganancia a el DOA de las ondas de sonido dado un DOA conocido. Como resultado, todas las direcciones, excepto el DOA, experimentan interferencias destructivas. Hay varias formas y tipos de arreglos de micrófonos, y cada micrófono puede ofrecer una ganancia direccional y una respuesta de frecuencia diferentes.

Delay and Sum se basa en la suposición de una fuente de campo lejano, en la que se asume que la fuente de sonido se encuentra a una distancia (D) por lo menos 10 veces mayor a la distancia que separa a los N micrófonos (d), es decir, $D >> d$. Las señales captadas por cada micrófono se representan mediante $x(n,t)$, donde n es el índice del micrófono y t el tiempo. A continuación, las señales se retrasan con un retardo de τ_n, donde τ_n es el retardo aplicado a cada micrófono y n es el índice de micrófono. Las señales retardadas se suman para formar la señal de salida $y(t)$:

$$y(t) = \sum_{n}^{N} x(n, t - \tau_n) \tag{2.8}$$

El retardo τ_n se elige para alinear la fase de las señales de cada micrófono en la dirección

de la fuente deseada. En el caso de una fuente de campo lejano y un arreglo de micrófonos lineal, el retardo τ_n viene dado por la ecuación:

$$\tau_n = \frac{d}{c}(sen\theta) \tag{2.9}$$

Donde d es la separación entre micrófonos del arreglo, c es la velocidad del sonido, θ es el DOA.

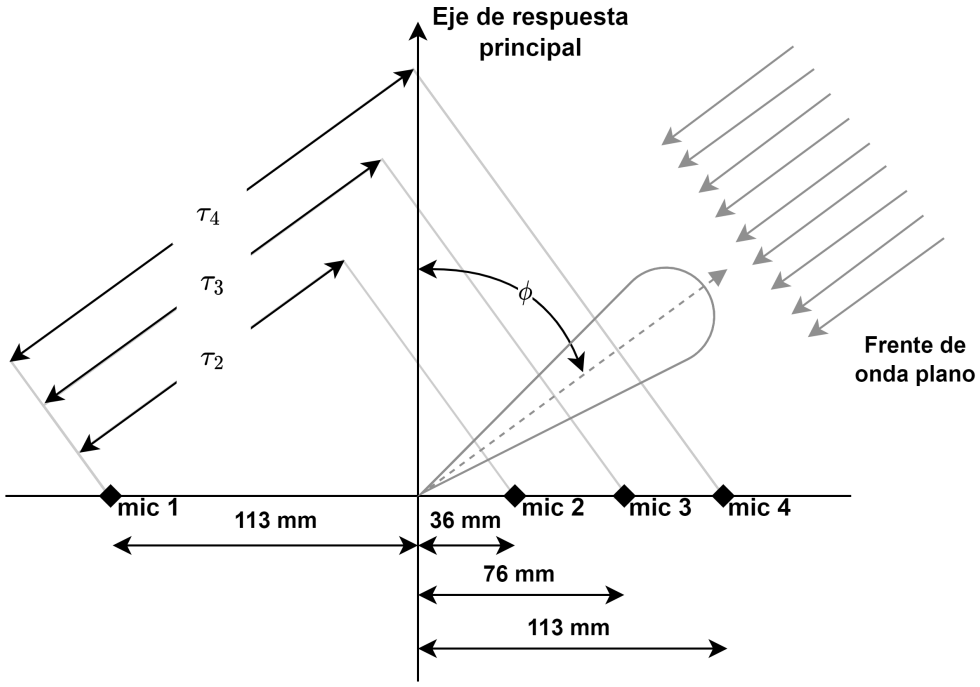


Figura 2.8: Diagrama de *beamforming* para geometría de Microsoft Kinect.

Una de las principales ventajas de el *beamforming* por *Delay and Sum* es su capacidad para proporcionar un alto nivel de directividad en la dirección deseada, suprimiendo al mismo tiempo las señales que llegan de otras direcciones. Esto se consigue mediante la interferencia constructiva y destructiva de las señales de cada micrófono del arreglo.

En general, *Delay and Sum* es un método sencillo y eficaz para el filtrado espacial en matrices de micrófonos, y se utiliza ampliamente en diversas aplicaciones como la mejora de la voz, la reducción del ruido y la localización de fuentes. Sin embargo, *Delay and Sum* también tiene sus limitaciones, siendo la principal su sensibilidad a la geometría del arreglo de micrófonos y a la posición relativa de estos, ya que estas afectan la ganancia que se obtiene para distintas frecuencias que posean una longitud de onda múltiplo de la distancia de separación de los micrófonos.

2.3.2.3. *Minimum Variance Distortionless Response* (MVDR)

MVDR es un tipo de *beamforming* que busca minimizar la varianza del ruido con la restricción de no-distorsión de la señal en la dirección deseada. Para lograr esto, el diseño del MVDR clásico [35] propone ponderar los canales con pesos dados por la solución al problema de optimización:

$$w_{\text{MVDR}}(f) = \arg\min_{w} w^{\mathsf{H}}(f)\mathbf{\Phi_{NN}}(f)w(f) \tag{2.10}$$

$$\text{s.t.} w(f)^{\mathsf{H}}\mathbf{d}(f) = 1 \tag{2.11}$$

Obteniéndose:

$$w^{H}(f) = \frac{\left(v^{H}(f)\Phi_{NN}^{-1}(f)\right)}{\left(v^{H}\Phi_{NN}^{-1}(f)v(f)\right)} \tag{2.12}$$

En MVDR, también se utiliza la fórmula 2.9 para calcular los desplazamientos de fase basándose en el vector de dirección. Existen dos alternativas para estimar los ángulos de incidencia: utilizando la información de las señales recibidas en los micrófonos o teniendo el *Angle of Incidence* (AOI) accesible (utilizando una cámara, por ejemplo).

La matriz de covarianza del ruido y el vector de dirección deben determinarse antes de calcular los pesos de *beamforming* MVDR. Analíticamente, se puede calcular la matriz de covarianza del ruido de manera analítica o mediante el uso de un detector de voz automático que permita detectar aquellos segmentos del audio que no presentan voz y por ende solo poseen ruido.

2.3.3. Modelamiento del canal acústico

El canal acústico se suele modelar como un sistema de una sola entrada y una sola salida. Además, se suele suponer que es lineal e invariable en el tiempo. La ecuación que describiría un canal como el anterior se muestra a continuación:

$$y(t) = h(t) * x(t) + n(t) \tag{2.13}$$

Donde $y(t)$ es la señal obtenida, $x(t)$ es la señal de audio original, $*$ significa convolución, $h(t)$ es la respuesta al impulso del sistema y $n(t)$ representa usualmente ruido blanco gaussiano [36].

Sin embargo, en una situación real de HRI se tiene que existe no sólo el emisor de audio objetivo, sino que también distintos emisores de sonidos con correlación distinta a cero, es decir, que no son aleatorios ni tienen uniformemente distribuida su potencia a lo largo de todo el espectro de frecuencias. Además, para el caso de que el receptor esté montado en un robot, existen ruidos internos provenientes de los motores, ventiladores y cualquier pieza móvil del robot [31]. Entonces, se propone el uso de un modelo que incorpore la interacción de los ruidos externos al robot con el canal acústico y, por otra parte, incluya de forma aditiva los ruidos internos del robot.

$$y(t) = h^{v}(t) * x(t) + n^{r.i.}(t) + \sum_{i=1}^{N_{ruidos}} h_i^{r.e.}(t) * n_i^{r.e.}(t) \tag{2.14}$$

Donde h^v es la respuesta al impulso del canal acústico entre la fuente de voz y el receptor, $n^{r.i.}$ representa los ruidos internos del robot, $n_i^{r.e.}(t)$ es el iésimo ruido proveniente de una fuente externa, $h_i^{r.e.}$ es la respuesta al impulso del canal acústico entre la iésima fuente de ruido y el receptor.

Para esta tesis, se utiliza el término específico *Room Impulse Response* (RIR) para referirse a, como su nombre en inglés lo indica, la respuesta al impulso de una habitación. Evidentemente, la RIR depende de la la posición tanto de la fuente de sonido como del receptor. Además, dado un micrófono receptor *no omnidireccional* (que tiene ganancias distintas para cada dirección) entonces se tiene que la RIR también depende del sentido al que apunta el patrón de recepción, puesto que las ondas directas y reflejadas son ponderadas de forma diferente según la orientación del micrófono.

2.3.3.1. Estimación de RIRs: método exponential swept-sine

Dado un sistema como en 2.13, obtener la respuesta al impulso de forma precisa no es trivial, esto debido a que el canal acústico es ligeramente no lineal. Sin embargo, Farinha [36] escoge de forma inteligente la señal de excitación y logra sobreponerse a estas no linealidades con buena presición. El método de *exponential swept-sine* consiste en reproducir desde un punto de emisión una señal sinusoidal que va cambiando su frecuencia de forma exponencial, barriendo todo el espectro de frecuencias que permita el parlante. La función de excitación está dada por:

$$x(t) = \sin\left[K \cdot \left(e^{\frac{t}{L}} - 1\right)\right] \tag{2.15}$$

Donde los K y L se calculan de la siguiente manera:

$$K = \frac{\mathrm{T} \cdot f_1}{\ln\left(\frac{f_2}{f_1}\right)} \tag{2.16}$$

$$L = \frac{\mathrm{T}}{\ln\left(\frac{f_2}{f_1}\right)} \tag{2.17}$$

Esta función de excitación $x(t)$ se eligió de tal manera que se cumpla:

$$x(t) * x(-t) = \delta(t) \tag{2.18}$$

Luego, para obtener $h(t)$ solo es necesario realizar:

$$h(t) = y(t) * x(-t) \tag{2.19}$$

Para el caso de estudio de esta tesis, los micrófonos de la Microsoft Kinect tienen una frecuencia de muestreo de 16kHz, por ende la máxima frecuencia que puede detectar es de 8kHz. En la Figura 2.9 se muestra el espectrograma de la señal sinusoidal con una frecuencia que varía exponencialmente hasta los 8kHz.

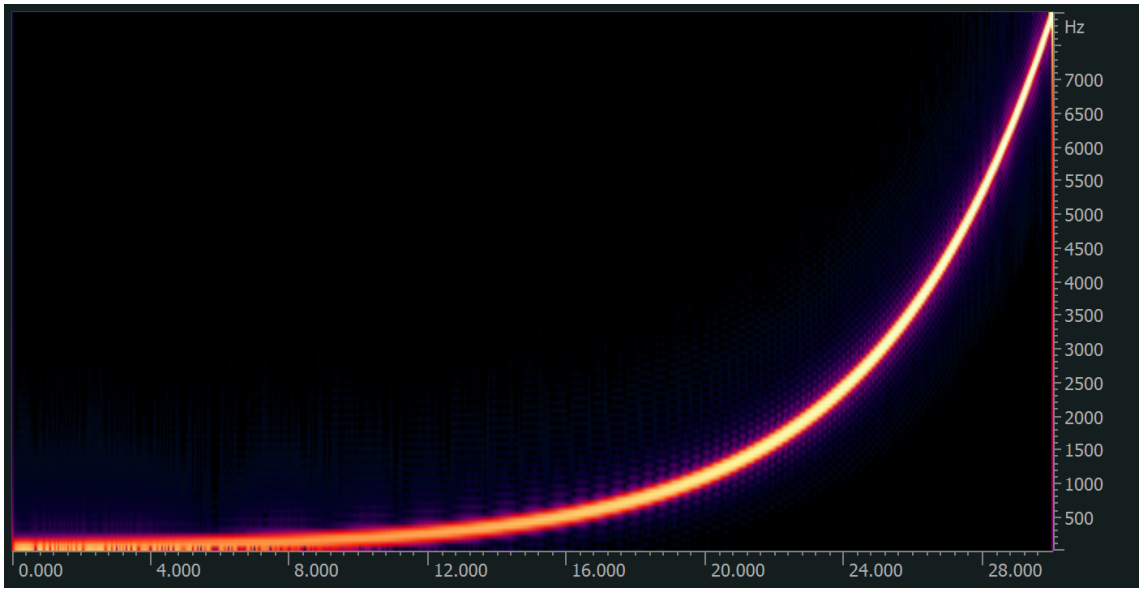


Figura 2.9: Espectrograma de la señal sinusoidal que varía su frecuencia de forma exponencial a través del tiempo.

Capítulo 3

Sistema de reconocimiento de emociones utilizando voz en HRI

3.1. Plataforma robótica y grabación de base de datos

La base de datos utilizada es el corpus MSP-Podcast (versión 1.9) del Laboratorio de Procesamiento de Señales Multimodales de la Universidad de Texas, Dallas. Este corpus es el mayor conjunto de datos emocionales naturalistas de la comunidad [37]. Cuenta con 86389 turnos de habla y, por tanto, acumula 137 horas de habla intermitente. Cada turno de habla tiene etiquetas emocionales que utilizan descriptores basados en atributos (*valence*, *arousal* y *dominance*) y etiquetas categóricas (felicidad, sorpresa, desprecio, neutro, ira, miedo, asco, tristeza y otros) que se registraron mediante *crowdsourcing*. Esta base de datos, a diferencia de la mayoría de las demás [38] [39] [40], contiene fragmentos de audio no actuado, en entornos de habla normales, por lo que se ajusta mejor al mundo real.

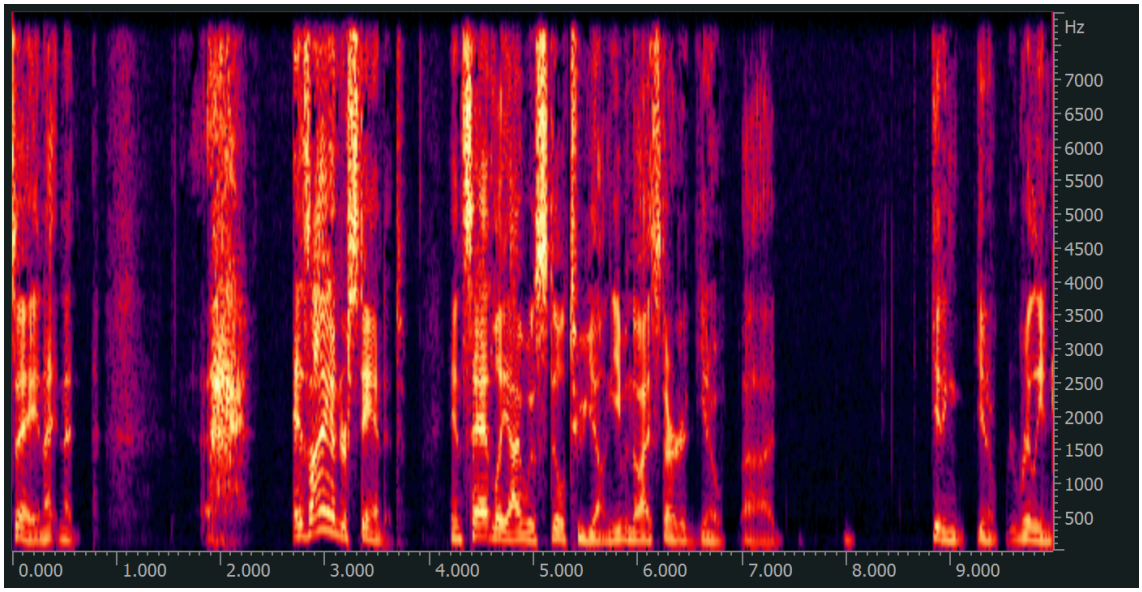


Figura 3.1: Ejemplo MSP-Podcast: espectrograma de la señal ***MSP-0160_0178.wav***.

Para el *testbed*, se elige grabar una partición de prueba del corpus en complejos escenarios reales de HRI que se explicarán más adelante. Esta partición de prueba tiene 21560 turnos de habla y acumula más de 32 horas de audio.

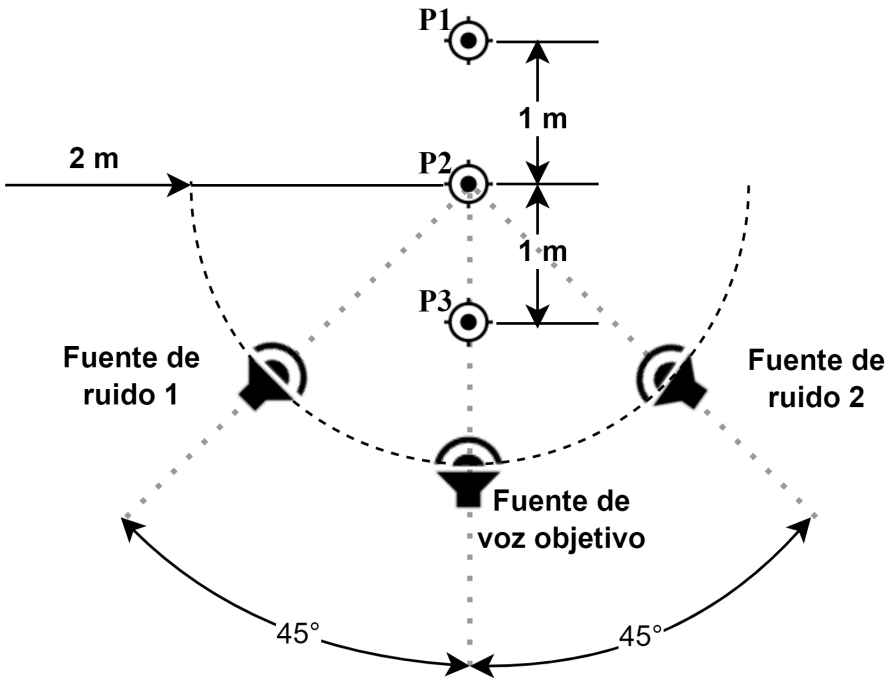


Figura 3.2: Configuración espacial del ambiente de prueba.

Para la configuración de prueba se utiliza un robot PR2 equipado con un sensor Kinect de Microsoft Xbox 360 montado en la parte superior de su cabeza. Como se muestra en la Figura 3.2, se usa una fuente de voz y dos fuentes de ruido, cada una situada a 2 m del punto P2. Las fuentes de ruido están situadas a 45º a cada lado de la fuente del habla. Se elige un *Signal-to-Noise Ratio* (SNR) de 5 dB medido desde el punto P2.

(a) Perspectiva trasera.

(b) Perspectiva frontal.

Figura 3.3: Fotografías de la configuración de prueba.

3.1.1. Escenario HRI estático

Para el escenario de grabación estático, el robot se sitúa manualmente de tal forma que el centro de la Microsoft Kinect montada en su cabeza se encuentre en el punto P1 ilustrado en la Figura 3.2. Además, mediante el uso de un programa preexistente que utiliza el entorno de trabajo de *Robot Operating System* (ROS) se fija la cabeza del robot para estar con un ángulo de rotación tanto vertical como horizontal de 0º.

3.1.2. Escenario dinámico

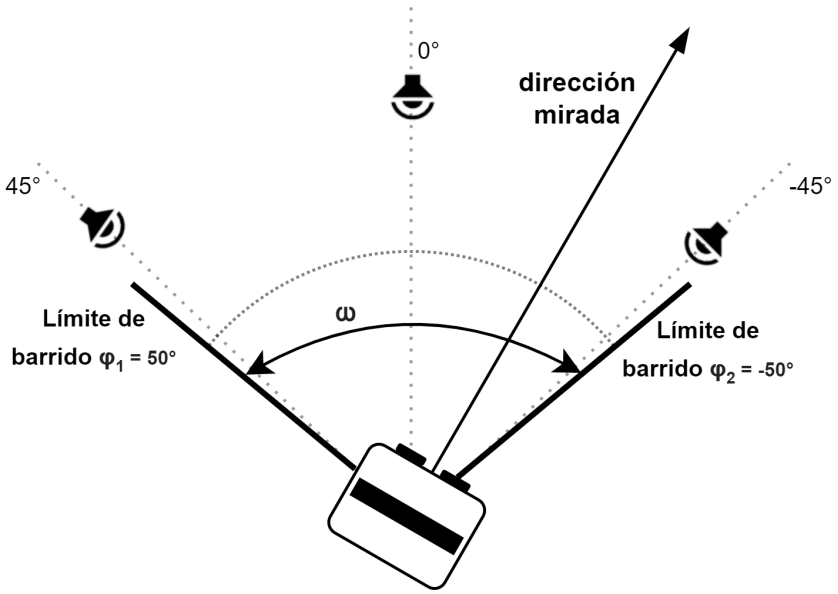


Figura 3.4: Configuración espacial del ambiente de prueba en *Sala de reuniones*.

En el escenario dinámico, PR2 se mueve entre P1 y P3 a una velocidad de 0,45 m/s. Además, la cabeza del robot se mueve periódicamente entre 50º y -50º con una velocidad angular constante de 0,56 rad/s, cambiando el objetivo visual del robot como se muestra en la Figura 3.4.

3.2. Sistema de SER en HRI propuesto

Se ha comprobado que en un escenario HRI el robot es capaz de utilizar sensores como cámaras para determinar la posición del usuario y, por tanto, tener un conocimiento casi inequívoco del ángulo de incidencia de la onda sonora en el arreglo de micrófonos [33]. Por lo tanto, la propuesta de este trabajo es el uso de *beamformers* que utilicen la información de la posición en tiempo real del hablante para realizar un filtrado espacial en esa dirección en cada ventana de tiempo y así conseguir mejores resultados en SER continuo para entornos complejos de interacción humano-robot.

Además, debido a que el robot debe realizar sus tareas en ambientes reales con presencia de ruido, reflexiones acústicas y dinamismo, se busca que los audios con los que se entrena el modelo de reconocimiento de emociones sean lo más cercanos a aquellos que serán procesados en el escenario real. Para esto, se busca modelar el canal acústico utilizando respuestas impulsivas obtenidas de forma experimental en el mismo ambiente de prueba final. La Figura 3.5 muestra el diagrama de flujo de la propuesta.

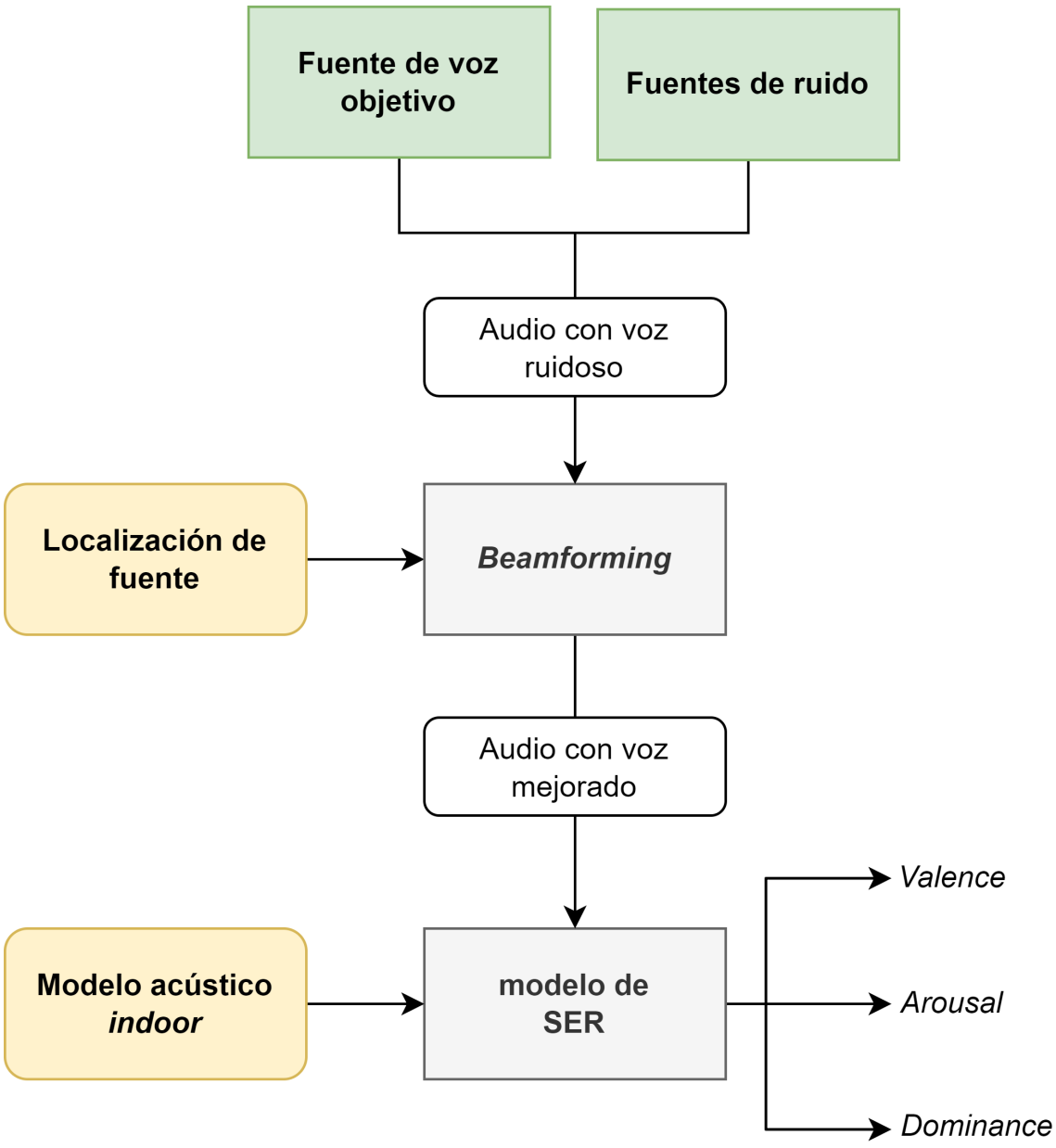


Figura 3.5: Diagrama del sistema de SER en HRI propuesto.

3.2.1. Modelamiento del canal de voz *indoor*

Con el fin de modelar el canal acústico indoor, se busca calcular las RIRs de forma experimental. Para esto se aplica el método de *swept-sine* detallado en la sección 2.3.3.1.

Dado que se requiere una representación robusta del canal acústico en la *sala de reuniones*, se calculan en total 63 RIRs distintas para cada uno de los cuatro micrófonos de la Microsoft Kinect.

Las 63 RIRs anteriores están divididas en tres posiciones y 21 ángulos distintos de orientación de la cabeza de PR2 por posición. Las tres posiciones en las que se calculó de forma

experimental son P1, P2 y P3 (Figura 3.2), y para cada posición del robot se orientó la cabeza en 21 ángulos diferentes con respecto a la fuente. El ángulo de la cabeza varia de -50º a 50º en pasos de 5º. El ángulo de 0º corresponde a la cabeza de PR2 orientada hacia el parlante que reproduce la señal de voz.

Tal como lo indica el método de *swept-sine*, el audio a reproducir para calcular las RIRs corresponde a una sinusoidal que realiza un barrido de frecuencia exponencial que recorre desde los 64 Hz a 8 kHz.

Una vez calculadas las 63 RIRs para el parlante en la posición en que se reproduce la voz, se repite el mismo procedimiento otras dos veces, con los parlantes ubicados en la posiciones de *ruido 1* y *ruido 2*. de forma de modelar por completo el ambiente real.

3.2.2. Creación de base de datos simulada

Para la generación de señales simuladas se utiliza la siguiente técnica:

- Un 25 % de los datos de cada partición (entrenamiento, validación y prueba) son convolucionados con la RIR de speech percibida por robot en P1 con la cabeza apuntando a 0º.
- El 75 % restante de las señales de cada partición es convolucionado de forma equitativa con las 62 RIRs restantes de *speech* y posteriormente sumadas a segmentos de ruidos del mismo largo con un SNR de entre 10dB y 20dB.

Estos ruidos, a su vez, fueron simulados de la siguiente forma: se utilizan segmentos de una librería de ruidos ambientales reales los cuales son convolucionados con la RIR para *fuente de ruido 1* y *fuente de ruido 2* correspondiente a la misma posición del robot de la señal de *speech* a la que están siendo sumados; al resultado de esta convolución se le suma de forma lineal el ruido real del robot PR2 con un SNR de entre -5 dB y 5 db.

3.2.3. Aplicación de técnicas de *beamforming*

3.2.3.1. Implementación de D&S

A continuación se muestra el pseudo código implementado para realizar D&S a nivel de *frame*.

Algoritmo 1 D&S utilizado

Require: $x(1,t), x(2,t), x(3,t), x(4,t)$ ▷ Señales de los 4 mic.
Require: $\theta = [\theta_1, \theta_2, ..., \theta_M]$ ▷ DOA para cada frame
Require: $d = [d_{1,2}, d_{1,3}, d_{1,4}]$ ▷ distancias de mics con el mic. ref.
  for m = 1:M **do**
    $[\tau_1, \tau_2, \tau_3, \tau_4](m) = CalcularTau(\theta(m), d)$ ▷ Se calculan los retardos por frame
    $t_{ini} = (m-1) * largoFrame$
    $t_{fin} = m * largoFrame$
    for $k = t_{ini} : t_{fin}$ **do**
      $y(k) = \sum_{c=1}^{4} x(c, k - \tau_c(m))$ ▷ Se aplica D&S
    end for
  end for

3.2.3.2. Implementación de MVDR

Como menciona en el Marco Teórico, para aplicar MVDR clásico se asume la estacionariedad de la varianza del ruido a lo largo de todo el audio. Sin embargo, debido a que en esta tesis se estudia un caso de Recorded dinámico, la estacionariedad de la varianza del ruido a lo largo de todo el audio no puede ser asumida. Es por esto que mediante el uso de un *Voice Activity Detector* (VAD), se segmenta el audio en segmentos con voz y sin voz, para luego procesar los *segmentos de voz* utilizando una interpolación de Φ_{NN} calculada con los *segmentos sin voz* colindantes.

Algoritmo 2 MVDR utilizado

Require: $x(1,m,t), x(2,m,t), x(3,m,t), x(4,m,t)$ ▷ Señales de los 4 mic.
Require: $\theta = [\theta_1, \theta_2, ..., \theta_M]$ ▷ DOA para cada frame
Require: $d = [d_{1,2}, d_{1,3}, d_{1,4}]$ ▷ distancias de mics con el mic. ref.
Sea $IniNoVoz_j$ el índice del frame donde inicia el j-ésimo segmento sin voz del audio y $FinVoz_j$ es el índice del frame donde termina el j-ésimo segmento sin voz del audio:
Require: $ListaNoVoz = [(IniNoVoz_1, FinNoVoz_1), ..., (IniNoVoz_J, FinNoVoz_J)]$
$x(c,m,t) \rightarrow X(c,m,omega)$ ▷ Se aplica DFT siendo m es el índice de frame
for $j = 1 : J$ **do**
  for $m = IniNoVoz_j : FinNoVoz_j$ **do**
    for $\omega = 1 : \Omega$ **do**
      $N(\omega) = [X(1,m,\omega), X(2,m,\omega), X(3,m,\omega), X(4,m,\omega)]$
      $\Phi_{NN}(j,\omega) += N(\omega)N^H(\omega)$
    end for
  end for
  $\Phi_{NN} /= (FinNoVoz_j - IniNoVoz_j)$
end for
for m = 1:M **do**
  $[\tau_1, \tau_2, \tau_3, \tau_4](m) = CalcularTau(\theta(m), d)$ ▷ Se calculan los retardos por frame
  $v(m,\omega) = \left[e^{-j\omega\tau_1(m)}, e^{-j\omega\tau_2(m)}, \ldots, e^{-j\omega\tau_4(m)}\right]^T$
  $\Phi_{NN}Actual = \begin{cases} \text{si es un frame de ruido, } \Phi_{NN}(j_{correspondiente}, \omega) \\ \text{si es de voz, interpolación}(\Phi_{NN}(j_{ant}, \omega), \Phi_{NN}(j_{sig}, \omega)) \end{cases}$
  $w^H(m,\omega) = \frac{\left(v^H(k,\omega)(\Phi_{NN}Actual)^{-1}(\omega)\right)}{(v^H(m,\omega)(\Phi_{NN}Actual)^{-1}(\omega)v(m,\omega))}$
  $Y(m,\omega) = w^H(m,\omega) \begin{bmatrix} (X(1,m,\omega) \\ \ldots \\ X(4,m,\omega) \end{bmatrix}$
end for
$Y(m,\omega) \rightarrow X(m,t)$ ▷ Se aplica transformada inversa de Fourier (IDFT)

3.2.4. Módulo de SER

Para el bloque de SER de la propuesta, se eligen dos modelos utilizados de forma reiterativa en la literatura: *ladder network* y Wav2vec 2.0. El primero, un método más tradicional, requiere que se extraigan *features* del audio antes de procesarlo. En cambio, el segundo, recibe el audio en *bruto* y solo requiere que este tenga media cero y desviación estándar unitaria.

3.2.4.1. Arquitectura del modelo Ladder Network utilizado

En la Figura 3.6 se muestra la arquitectura de la *ladder network* utilizada para los experimentos. La entrada del *encoder* es un vector de características de dimensión 6373. El *encoder* consiste en un MLP de dos capas ocultas con 256 neuronas cada una y una capa de salida de dimensión tres, es decir, una salida por dimensión emocional (*Arousal*, *Valence* y *Dominance*). Por otro lado, el *decoder* tiene las mismas capas, pero en orden inverso. Además, el *decoder* no solo recibe la salida de la capa anterior, sino también la salida con ruido gaussiano

de la capa correspondiente del *encoder*.

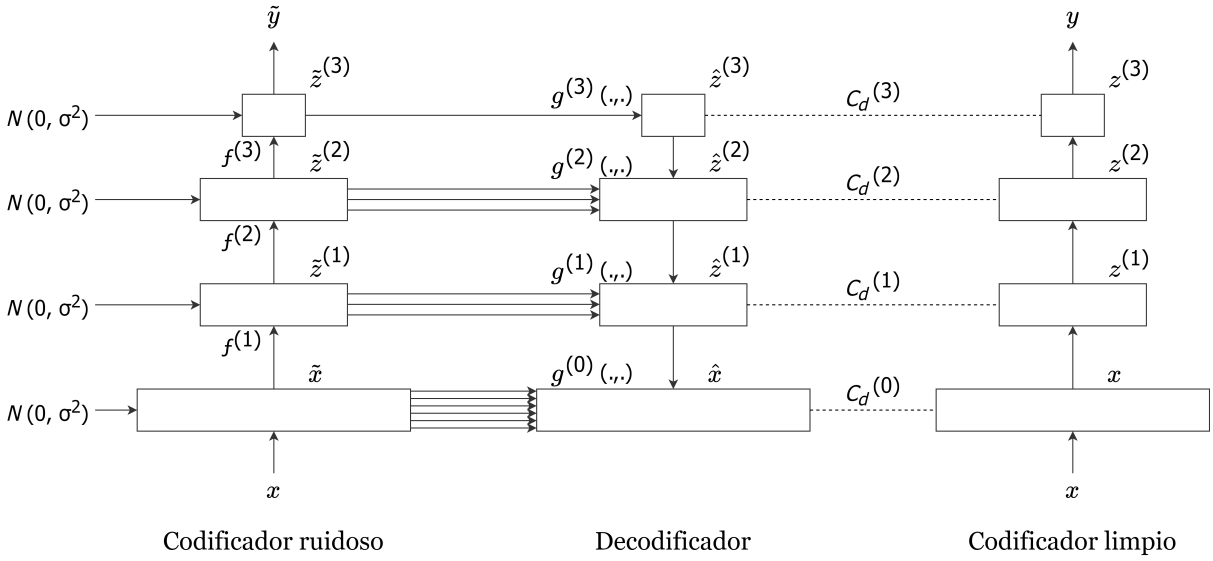


Figura 3.6: Configuración espacial del ambiente de prueba.

3.2.4.2. Arquitectura del modelo Wav2vec 2.0

En la Figura 3.7 se muestra el modelo de reconocimiento de emociones basado en Wav2vec utilizado para los experimentos. Este se puede dividir en cuatro componentes principales: el bloque extractor de características con capas convolucionales, el bloque de atención con *transformer*, la capa de agrupación (que aplica promedio global a lo largo del audio) y el bloque de regresión con capas ocultas que entrega la salida del modelo.

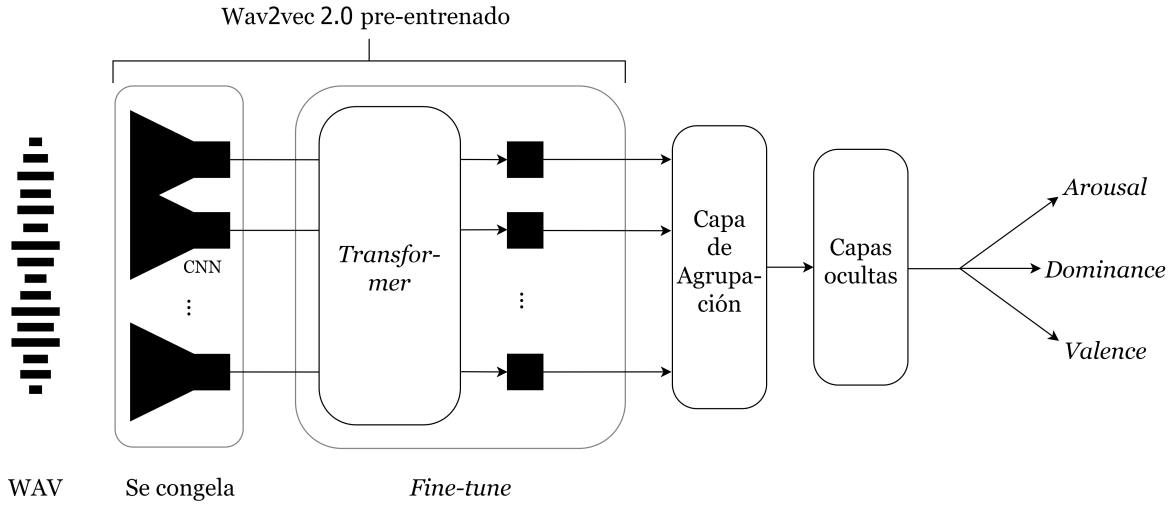


Figura 3.7: Módulo de SER con Wav2vec 2.0.

En la Figura 3.8 se muestra con mayor detalle la estructura del bloque de extracción de características. Se puede calcular la segunda dimensión de salida de cada capa convolucional en función de L_{audio} utilizando la fórmula 2.3.

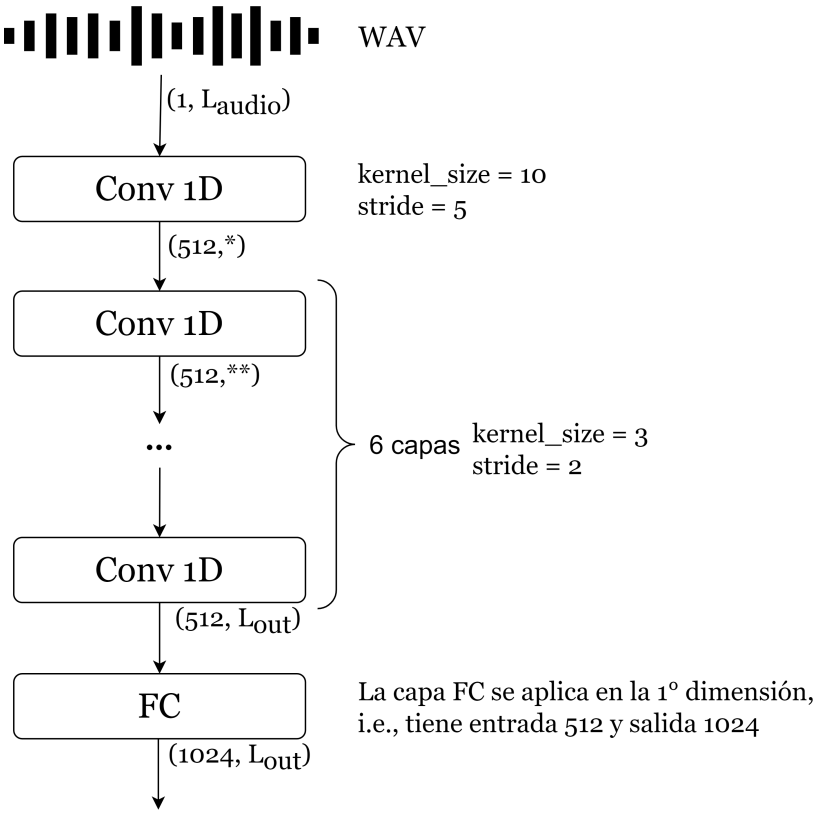


Figura 3.8: Bloque extractor de características con capas CNN de Wav2vec 2.0.

Continuando con el flujo del modelo basado en Wav2vec, en la Figura 3.9 se detalla las capas y dimensiones de las entradas y salidas de este. La capa atencional utiliza atención multicabezal, la cuál es descrita en la sección 2.2.3.6.

Posteriormente, la salida del *transformer* es pasada por la capa de agrupación, la que calcula un promedio lo largo de todo el audio para cada feature, dejando un vector de dimensión 1024. Por último, este array es propagado a través de las capas *fully conected* (ver Figura 3.10), para obtener el resultado de *Arousal*, *Valence* y *Dominance*.

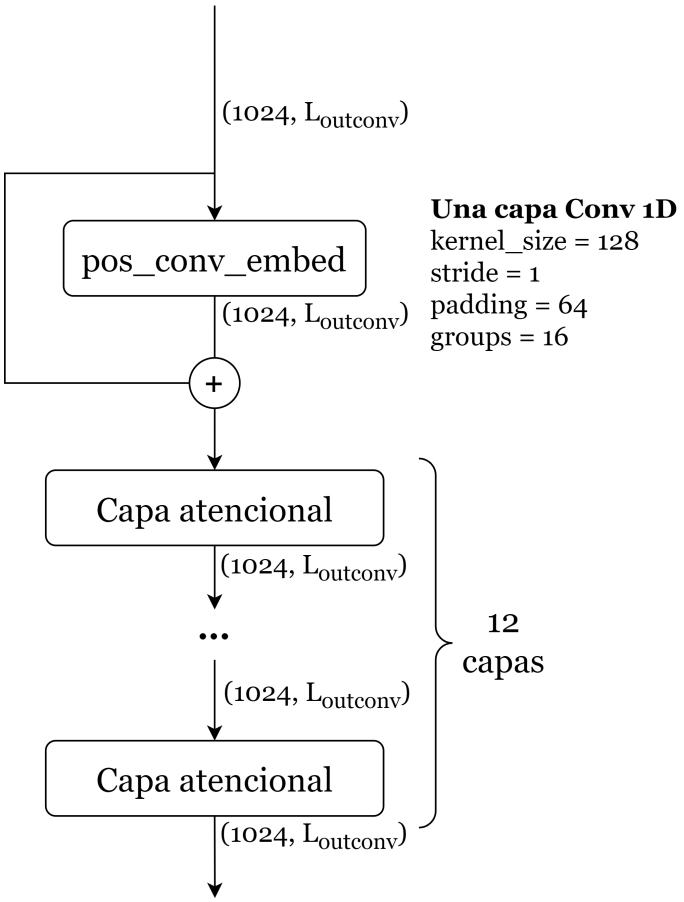


Figura 3.9: Bloque *transformer* de Wav2vec 2.0.

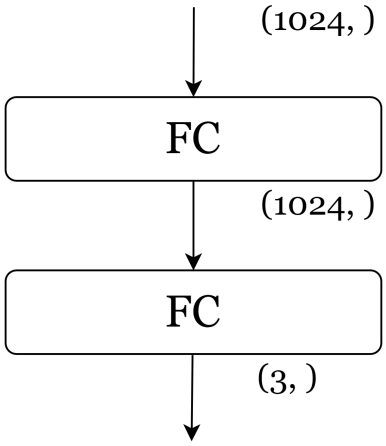


Figura 3.10: Capas ocultas con Wav2vec 2.0.

3.3. Descripción experimental

3.3.1. Descripción de bases de datos de entrenamiento

Para el entrenamiento de los modelos SER se utilizan dos tipos diferentes de bases de datos. El primer tipo consiste en el corpus original de MSP-Podcast. El segundo tipo consiste en los mismos audios, pero procesados por el modelo que simula el canal acústico real.

3.3.2. Descripción de bases de datos de prueba

Los modelos se prueban con los siguientes conjuntos de datos:

- Los audios de la partición de prueba del corpus MSP-Podcast (original).
- Los mismos audios, pero simulados utilizando lo descrito en la sección 3.2.2.
- Los mismos audios, pero re-grabados usando la plataforma robótica en el escenario estático descrito en la sección 3.1.1.
- Los mismos audios, pero re-grabados usando la plataforma robótica en el escenario dinámico descrito en la sección 3.1.1.

Para los últimos tres datasets, se estudia el uso de D&S y MVDR.

En al siguiente tabla se ilustra de forma ordenada las distintas configuraciones de conjuntos de prueba que se evalúan.

Tabla 3.1: Resumen de configuraciones.

Condición de base de datos	**Usa MVDR**	**Usa D&S**	**Nombre**
Original			Original
Simulada		✓	Sim+D&S
	✓		Sim+MVDR
HRI Estática			Recorded static
		✓	Recorded static+D&S
	✓		Recorded static+MVDR
Recorded dinámica			Recorded din
		✓	Recorded din+D&S
	✓		Recorded din+MVDR

3.3.3. Procedimiento de entrenamiento de Módulo de SER

3.3.3.1. Entrenamiento semi-supervisado de Ladder network

La extracción de características se realiza utilizando el *toolkit* openSMILE. Este es un programa de código abierto para la extracción de características de audio y la clasificación de señales de voz y música.

Como entrada a la red utilizamos el conjunto de 6373 *features* unificadas de [41], estos son High Level Descriptors (HLD), los cuales son independientes de la duración del audio.

En la siguiente ecuación se tiene la función de costo general de una *ladder network*:

$$C_{\text{Ladder}} = C_c + \sum_l \lambda_l C_d^{(l)} \tag{3.1}$$

Donde C_c es la función de costo asociada a la tarea supervisada, es decir, con ejemplos. Para el caso de este trabajo, la función de costo corresponde a:

$$C_{\text{Lad }+MTL} = \alpha C_{\text{aro}} + \beta C_{\text{val}} + (1-\alpha-\beta)C_{dom} + \sum_l \lambda_l C_d^{(l)} \tag{3.2}$$

3.3.3.2. Procedimiento de *Fine-tunning* de Wav2vec

Se utiliza el mejor modelo en SER continuo para MSP-Podcast encontrado en la investigación "*Dawn of the transformer era in speech emotion recognition: closing the valence gap*" [42]: un MLP sobre un modelo preentrenado Wav2vec 2.0-large-robust [43]. La conexión entre el transformador y el MLP consiste en aplicar *average pooling* sobre los estados ocultos de la última capa del transformador y hacerlos pasar por una capa totalmente conectada y una capa final de salida. Hay *dropout* antes de las dos capas de la cabeza. Antes del *fine-tunning*, se congelan los pesos de las capas de la CNN, pero se entrenan la sección correspondiente al transformer y el MLP. Esto se debe a que, según [44], se obtienen mejores resultados congelando las capas de extracción de características al realizar SER. El optimizador elegido es ADAM con una tasa de aprendizaje de 10^{-4} y un tamaño de *batch* de 32.

3.3.4. Métricas de rendimiento

3.3.4.1. *Signal-to-Noise Ratio* (SNR)

Como métrica del desempeño de los *beamformers* se utiliza el SNR, ya que esta medida permitirá comparar la relación entre la potencia de la señal de voz con la potencia del ruido presente en el audio. Se comparará el SNR de la señal grabada con la señal grabada después de aplicarle *beamforming*, siendo el resultado deseado un SNR de la señal obtenida más alto después del *beamformer*. En el contexto de esta tesis se considera como ruido toda señal o perturbación distinta a la señal de voz objetivo, independientemente de que estas sean interferencias intencionales (otras fuentes de sonido) o no.

3.3.4.2. *Concordance Correlation Coefficient* (CCC)

Como métrica de desempeño de reconocimiento de emociones continuo se utiliza el *Concordance Correlation Coefficient* (CCC). Como se explica en la sección 2.2.2.1, el CCC corresponde a una medida de la concordancia entre dos conjuntos de mediciones continuas, con un rango de valor del -1 al 1, donde cerca de 1 significa un fuerte acuerdo entre los conjuntos y cerca de -1 sugiere un grave desacuerdo. Sus características lo convierten en una sólida medida de la concordancia.

Capítulo 4

Resultados y discusión

4.1. Resultados de *beamforming* para HRI

En la Tabla 4.1 se muestran los resultados en términos de SNR del escenario estático para las técnicas de *beamforming*. Se observa que para estático tanto D&S como MVDR mejoran el caso que no utiliza *beamforming* en un 52.75 % y 71.25 %, respectivamente. El mayor incremento de SNR gracias al uso de MVDR con respecto al uso de D&S se condice con lo mostrado en la literatura en HRI [31].

Tabla 4.1: Resultados de *beamforming* para condición estática.

Test type	**SNR [dB]**
Recorded static	5.46
Recorded static + D&S	8.34
Recorded static + MVDR	9.35

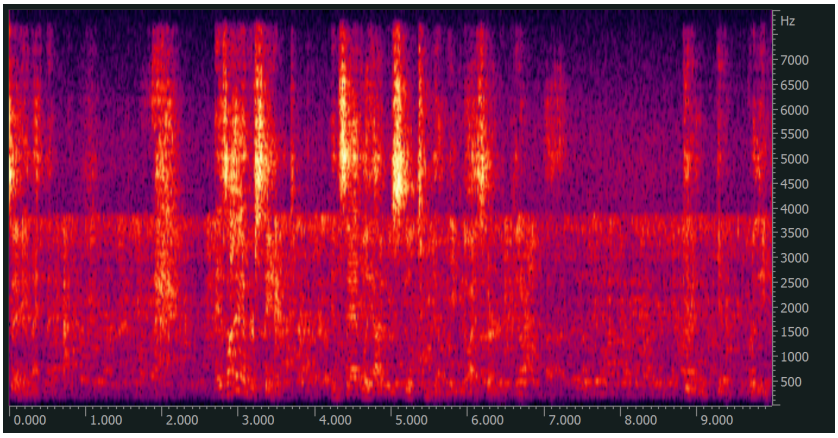
En la Tabla 4.2 se muestra el rendimiento de las técnicas de *beamforming* para el escenario dinámico. Se nota que tanto D&S como MVDR mejoran el caso que no utiliza *beamforming* en un 47.10 % y 90.16 %. Al igual que en el escenario estático, se obtiene mejores resultados en términos de SNR con MVDR.

Cabe destacar que en términos porcentuales, el incremento de SNR para D&S es mayor para el escenario estático, mientras que el incremento porcentual al usar MVDR es mayor en el casó dinámico. El peor rendimiento de D&S en el caso dinámico se puede deber a la rotación de la cabeza del robot, ya que esta genera una variación en el tiempo de la ganancia de los micrófonos en dirección a la señal de voz objetivo. Por otra parte, el mejor rendimiento del *beamforming* con MVDR en el escenario dinámico se puede deber a que este método logra ignorar los efectos negativos de la variación de la ganancia hacia la señal objetivo y aprovecha los momentos en que el robot está más próximo a la fuente objetivo de sonido.

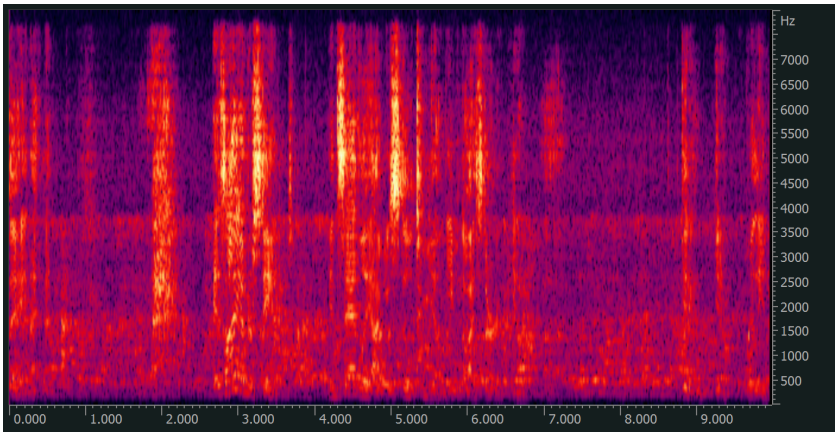
Tabla 4.2: Resultados de *beamforming* para condición dinámica.

Test type	**SNR [dB]**
Recorded din	5.69
Recorded din + D&S	8.37
Recorded din + MVDR	10.82

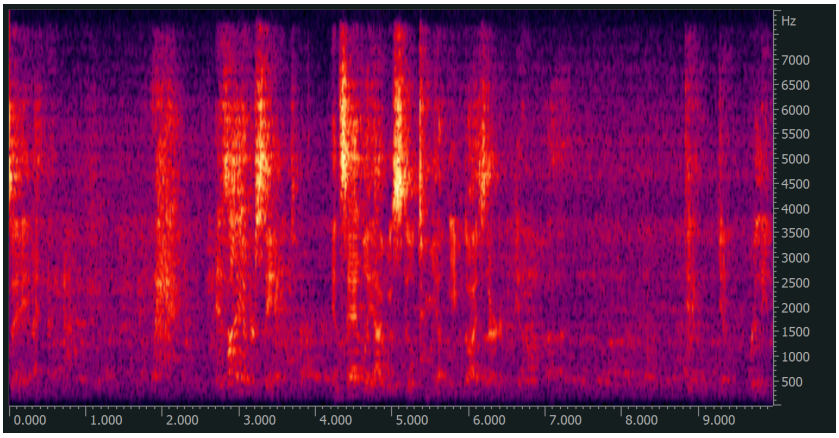
En las Figuras 4.1 y 4.2 se muestra los espectrogramas de un audio elegido al azar para los casos HRI, HRI+D&S y HRI+MVDR, de los escenarios estático y dinámico, respectivamente. Las lineas verticales más marcadas corresponden a vocalizaciones, mientras que las zonas horizontales más marcadas corresponden a una mayor presencia de ruido. En estas imágenes se puede apreciar de forma visual como tanto D&S y MVDR logran limpiar parcialmente los audios del ruido ambiental, lo cuál es consistente con los resultados de SNR obtenidos. Se observa que, particularmente para el audio mostrado en las imágenes, se tiene un mejor SNR original (señal sin *beamforming*) del caso dinámico, lo cuál probablemente se debe a que el robot se encontraba en un punto cercano a P3 de la Figura 3.2 mientras se desplazaba grabando.



(a) Antes de cualquier *beamforming*.

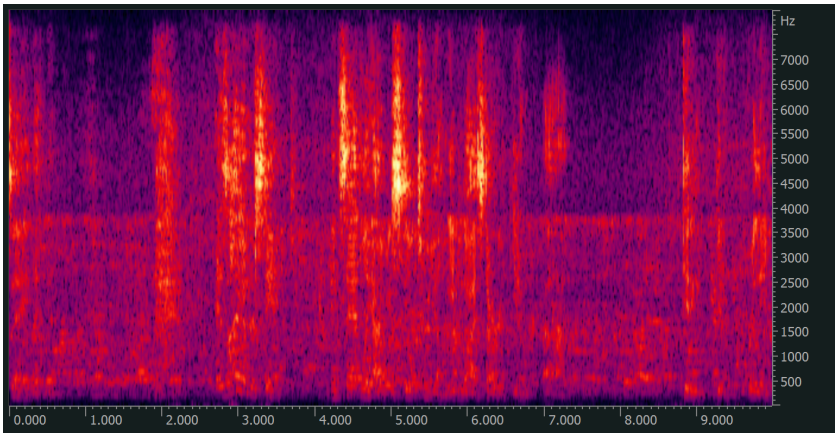


(b) Después de D&S.

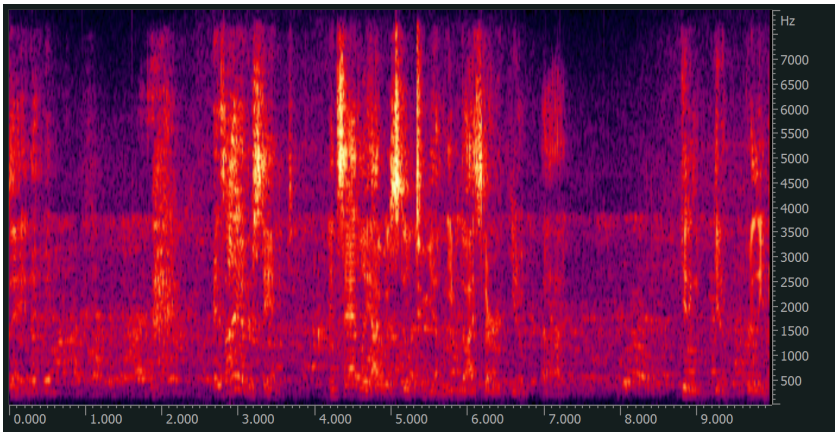


(c) Después de MVDR.

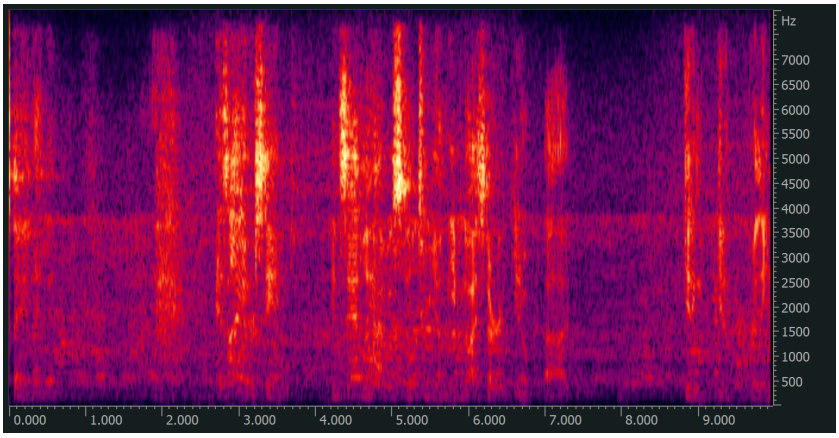
Figura 4.1: Ejemplo de espectrogramas obtenidos para la señal ***MSP-0160_0178.wav*** en escenario estático.



(a) Antes de cualquier *beamforming*.



(b) Después de D&S.



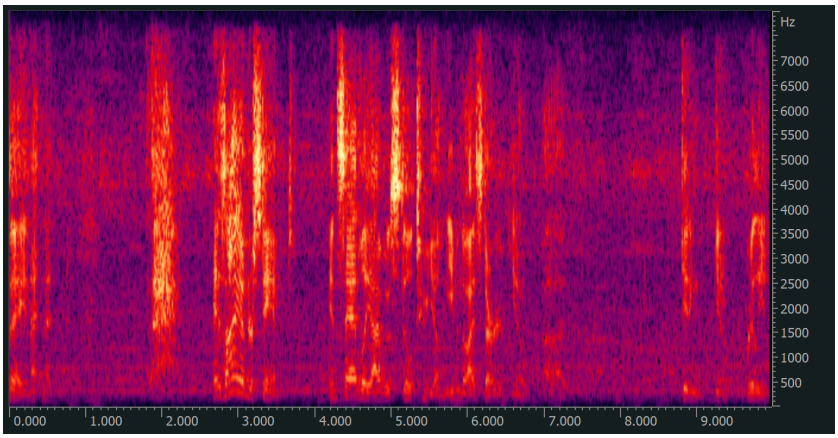
(c) Después de MVDR.

Figura 4.2: Ejemplo de espectrogramas obtenidos para señal ***MSP-0160_0178.wav*** en escenario dinámico.

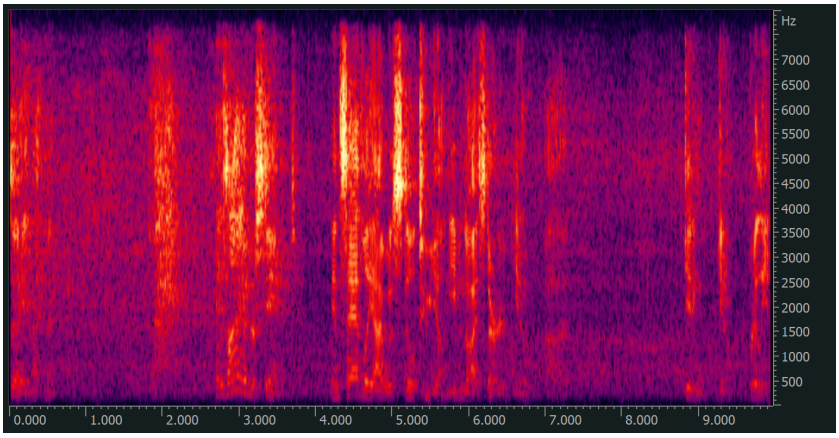
4.2. Resultados de simulación

En la Figura 4.3 se muestran los espectrogramas obtenidos tras utilizar el proceso de modelamiento descrito en la sección 3.2.1 con el mismo audio de ejemplo usado anteriormente. Mediante una inspección acústica cualitativa de audios al azar, el estudiante determina que

es imposible distinguir un audio real de uno simulado, ya que estos suenan naturales. Ahora bien, al realizar una inspección visual de los espectrogramas simulados, se logra observar que el ruido está distribuido de forma más homogénea a lo largo de todas las frecuencias en el caso de los audios simulados. Se concluye que para medir la eficacia de la simulación se deberá analizar los resultados de los modelos de SER entrenados con esta técnica.



(a) Después de D&S.



(b) Después de MVDR.

Figura 4.3: Ejemplo de espectrogramas obtenidos para señal ***MSP-0160_0178.wav*** en base de datos simulada.

4.3. Resultados de reconocimiento de emociones con voz

4.3.1. Entrenamiento con base de datos original y evaluación en base de datos HRI

La Tabla 4.3 muestra el *Concordance Correlation Coefficient* (CCC) obtenido con el sistema entrenado con el corpus original MSP-Podcast (original) para distintas configuraciones. En esta tabla, la presencia de ruido y los efectos del canal del habla provocan un deterioro considerable del rendimiento del modelo en todos los atributos emocionales.

Tabla 4.3: Resultados obtenidos con modelos entrenados con el dataset original, probados en condiciones estáticas.

Model	Test type	CCC Aro	CCC Dom	CCC Val
Ladder Network	Original	0.629	0.536	0.266
	Recorded static	0.175	0.0655	0.0732
	Recorded static + D&S	0.3428	0.2332	0.0789
	Recorded static + MVDR	0.3125	0.2507	0.1178
Wac2vec	Original	0.599	0.496	0.518
	Recorded static	0.4349	0.3559	0.263
	Recorded static + D&S	0.4782	0.3965	0.329
	Recorded static + MVDR	0.4147	0.3125	0.2886

La Figura 4.4 muestra una comparación visual del deterioro por el escenario real HRI y su posterior recuperación parcial al aplicar D&S y MVDR para los modelos SER, respectivamente. El modelo con mejor rendimiento en el escenario estático real es el que utiliza Wav2vec, esto podría ser debido al preentrenamiento que utiliza el modelo Wav2vec.

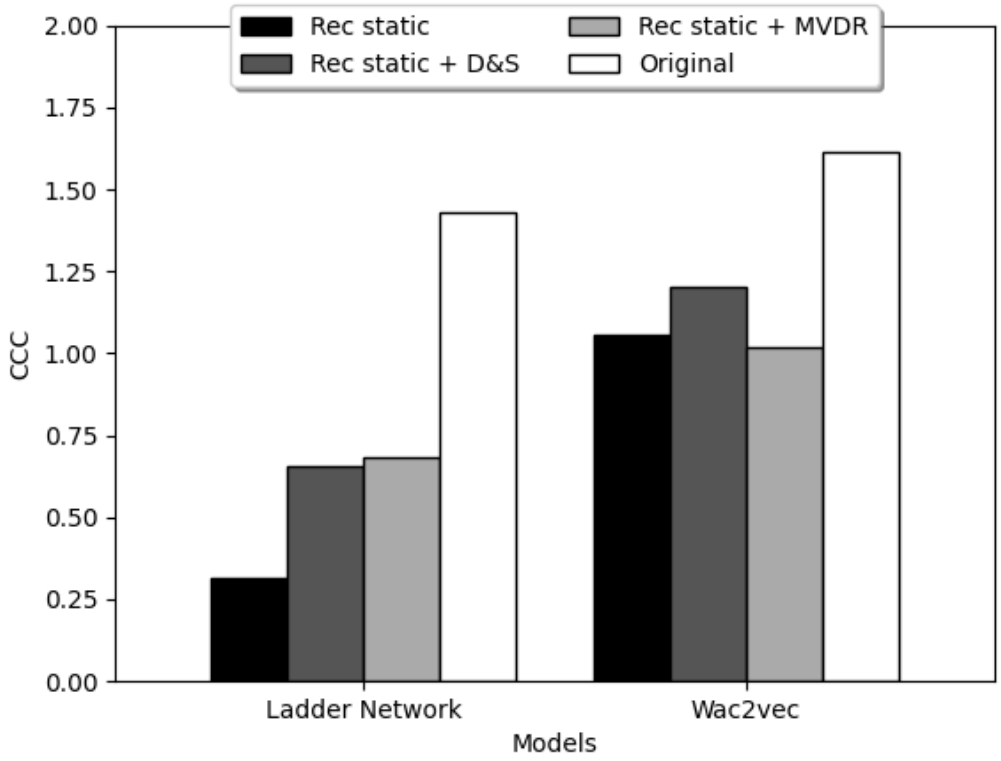


Figura 4.4: Suma de CCC para *valence*, *arousal* y *dominance* para diferentes modelos, probados en diferentes configuraciones en el escenario estático. Todos los modelos se han entrenado con datos originales.

En el caso de la red en escalera, el deterioro es del 78.08 %, en cambio, con el modelo Wav2vec el deterioro es sólo del 34.66 %.

La Tabla 4.4 muestra la CCC obtenida con el sistema entrenado con datos originales para distintas configuraciones de prueba en el escenario dinámico. Al igual que en el escenario estático, de esta tabla se desprende claramente que la presencia de ruido y el canal provocan un deterioro considerable del rendimiento del modelo.

Tabla 4.4: Resultados obtenidos con modelos entrenados con el *dataset* original, probados en condiciones dinámicas.

Model	**Test type**	**CCC Aro**	**CCC Dom**	**CCC Val**
Ladder Network	Original	0.629	0.536	0.266
	Recorded din	0.1978	0.0803	0.0883
	Recorded din + D&S	0.3652	0.2269	0.0783
	Recorded din + MVDR	0.3041	0.212	0.1035
Wac2vec	Original	0.599	0.496	0.518
	Recorded din	0.4342	0.353	0.2827
	Recorded din + D&S	0.4646	0.3835	0.3075
	Recorded din + MVDR	0.4092	0.3034	0.3143

En la Figura 4.5 se muestra una comparación visual del rendimiento de los modelos en el escenario dinámico real. Para ambas arquitecturas D&S tiene un mejor rendimiento que MVDR, esto podría ser debido a que la condición dinámica afecta más a la eficacia del *beamformer* MVDR, ya que asume estacionariedad de los ruidos. El modelo Wav2vec es el que presenta el mejor rendimiento en el escenario dinámico real en términos de CCC.

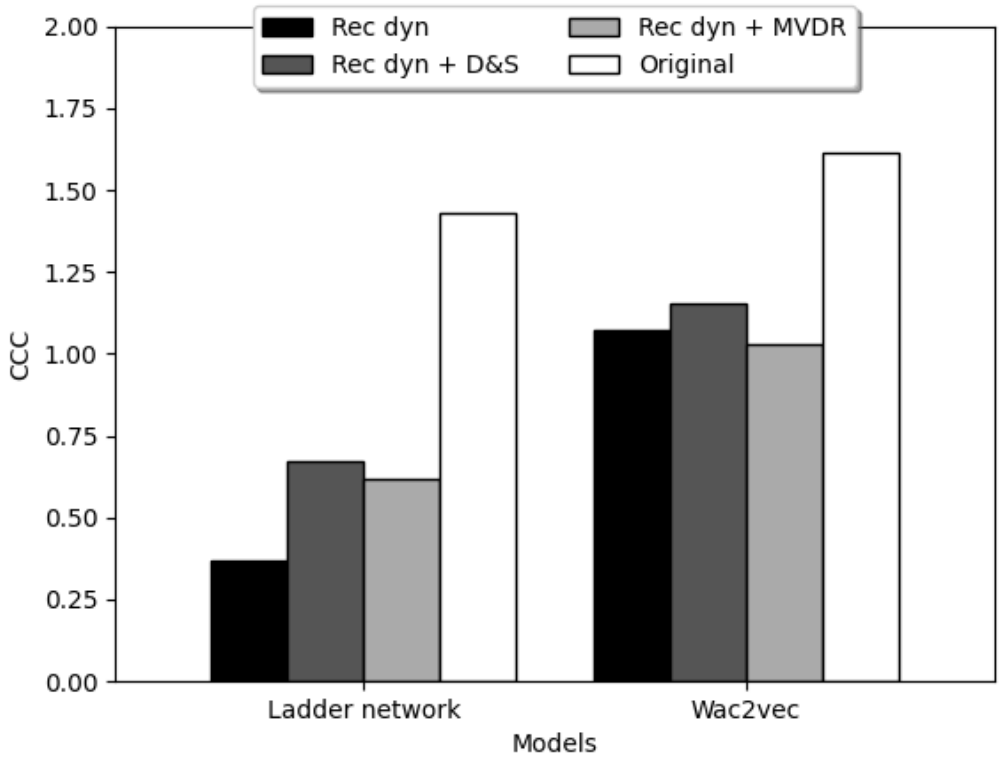


Figura 4.5: Suma de CCC para *valence*, *arousal* y *dominance* para diferentes modelos, probados en diferentes configuraciones en el escenario dinámico. Todos los modelos se han entrenado con datos originales.

4.3.2. Modelos entrenados con base de datos simulada y evaluados en base de datos simulada

La Tabla 4.5 muestra los resultados obtenidos con los modelos entrenados con el conjunto de datos de prueba simulado. Aunque sigue habiendo una degradación en comparación con los resultados obtenidos con el conjunto de datos de prueba original, esta degradación es inferior a la que se produce con los modelos entrenados con el conjunto de datos original y probados con los conjuntos de datos HRI reales utilizando los mismos formadores de haz.

Tabla 4.5: Resultados obtenidos con modelos entrenados y probados con datos simulados.

Model	**Train and test type**	**CCC Aro**	**CCC Dom**	**CCC Val**
Ladder Network	Sim + D&S	0.520	0.374	0.120
	Sim + MVDR	0.492	0.352	0.095
Wac2vec	Sim + D&S	0.529	0.442	0.345
	Sim + MVDR	0.508	0.404	0.216

En la Figura 4.6 puede verse que, tanto para las redes *ladder* como para las Wav2vec,

los resultados obtenidos con D&S son mejores que los obtenidos con MVDR. Esto significa que el artefacto que introduce MVDR no puede ignorarse incluso cuando se reentrenan los modelos.

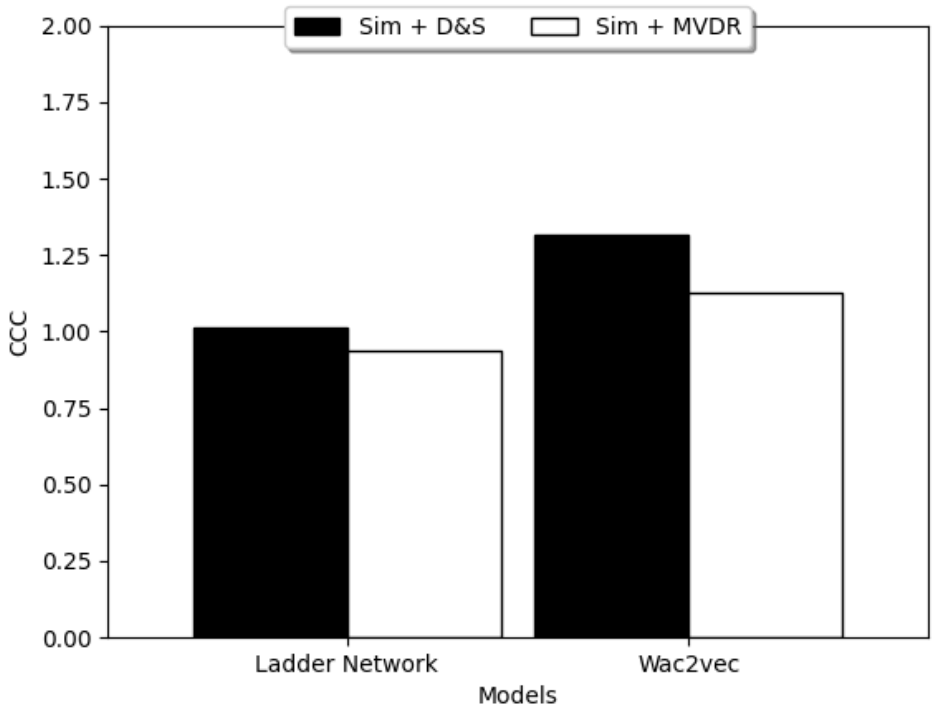


Figura 4.6: Suma de CCC para *valence*, *arousal* y *dominance* para diferentes modelos entrenados y evaluados con la base de datos simulada.

4.3.3. Modelos entrenados con base de datos simulada y evaluados en HRI real

La Tabla 4.6 muestra los resultados en términos de CCC obtenidos con modelos entrenados utilizando la base de datos simulada descrita en la sección 3.2.2 para la base de datos estática real. De esta tabla se desprende que utilizando datos que simulan el entorno real durante el entrenamiento, es posible aumentar el rendimiento de la red. Se observa que los resultados obtenidos con la base de datos estática real son similares a los obtenidos con la partición de prueba de la base de datos simulada, lo que demuestra que la simulación es representativa de los datos reales.

Tabla 4.6: Resultados obtenidos con modelos entrenados con datos simulados y probados en condiciones estáticas.

Model	**Train type**	**Test type**	**CCC**		
			Aro	**Dom**	**Val**
LN	Sim + D&S	Rec static + D&S	0.426	0.3166	0.093
	Sim + MVDR	Rec static + MVDR	0.437	0.342	0.100
Wac2vec	Sim + D&S	Rec static + D&S	0.5411	0.466	0.363
	Sim + MVDR	Rec static + MVDR	0.552	0.394	0.263

La Figura 4.7 muestra el rendimiento de los modelos entrenados con el conjunto de datos simulado en el escenario estático real, donde se puede observar que para la *ladder network* MVDR supera ligeramente a D&S, mientras que ocurre lo contrario para el modelo Wav2vec. De nuevo, la arquitectura Wav2vec supera el rendimiento de la *ladder network*.

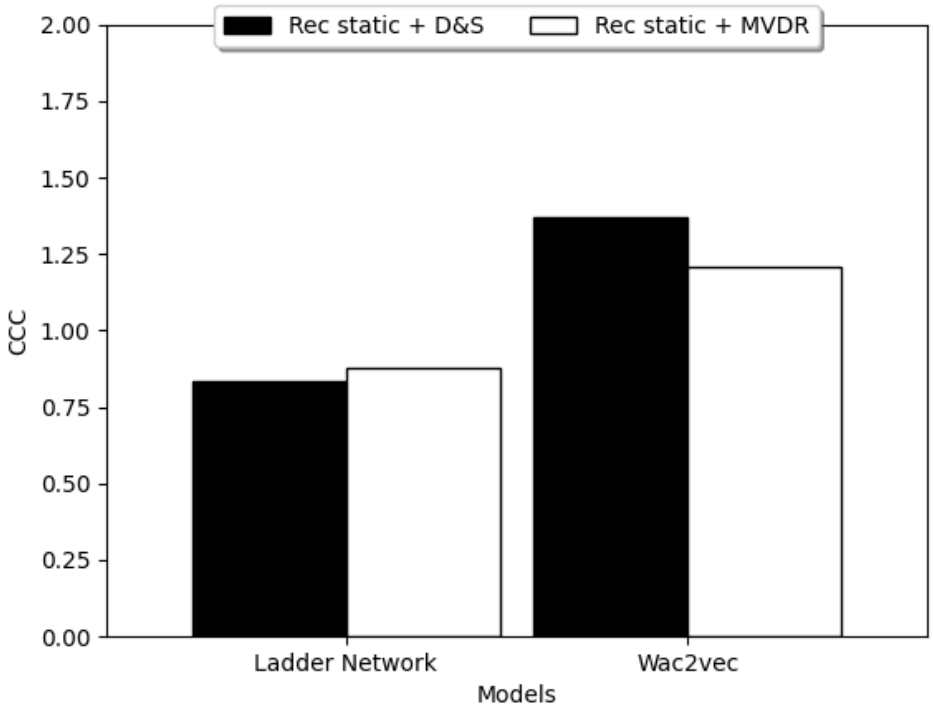


Figura 4.7: Suma de CCC para *valence*, *arousal* y *dominance* para diferentes modelos entrenados y evaluados en la base de datos HRI estática.

La Tabla 4.7 muestra los resultados en términos de CCC obtenidos con modelos que utilizan la base de datos simulada para la base de datos dinámica real. Al igual que en el caso estático, el rendimiento del modelo aumenta al entrenarlo con datos que simulan un entorno real. Como se observó en la base de datos estática, los resultados obtenidos con la base de datos dinámica real son similares a los obtenidos con la partición de prueba de la base de datos simulada (Tabla 4.6), lo que confirma una vez más que la simulación realizada representa correctamente los datos reales.

Tabla 4.7: Resultados obtenidos con modelos entrenados con datos simulados y probados en condiciones dinámicas.

Model	**Train type**	**Test type**	**CCC**		
			Aro	**Dom**	**Val**
LN	Sim + D&S	Rec dyn + D&S	0.4988	0.3647	0.112
	Sim + MVDR	Rec dyn + MVDR	0.4658	0.357	0.0995
Wac2vec	Sim + D&S	Rec dyn + D&S	0.5404	0.4592	0.3323
	Sim + MVDR	Rec dyn + MVDR	0.5576	0.4064	0.2805

La Figura 4.8 muestra el rendimiento de los modelos entrenados con el conjunto de datos simulado en el entorno dinámico real, donde D&S supera ligeramente a MVDR tanto para la *ladder network* como para los modelos Wav2vec. Una vez más, el modelo Wav2vec supera a aquel que utiliza *ladder network*.

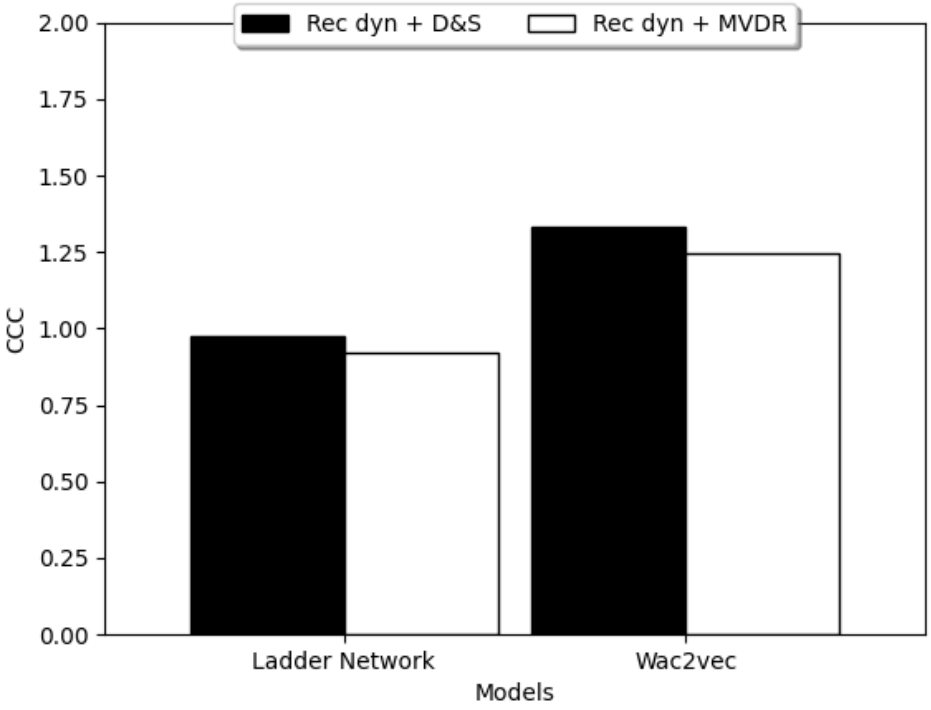


Figura 4.8: Suma de CCC para *valence*, *arousal* y *dominance* para diferentes modelos entrenados y evaluados en la base de datos Recorded dinámica.

4.3.4. Discusión

Según la Tabla 4.3, la mayor degradación en CCC *Arousal*, CCC *Dominance* y CCC *Valence* en comparación con la base de testeo *Original* se obtuvo con *Recorded Static* con *Ladder Network*. La degradación con Wav2vec fue mucho menor. Esto debe deberse al hecho de que este modelo se preentrenó utilizando 2.000 horas de datos telefónicos ruidosos [45]. Los esquemas de *beamforming* D&S y MVDR aumentan el SNR y disminuyen la degradación en

CCC *Arousal*, CCC *Dominance* y CCC *Valence* en comparación utilizando *Ladder Newtwork*. Como puede verse en la Figura 4.4, al compararse con *Original*, el uso de D&S y MVDR conducen a un aumento en la suma de CCC igual al 108.77 % y 117.09 %, respectivamente, cuando se utiliza *Ladder Network*. Sorprendentemente, sólo D&S pudo aumentar la suma de CCC cuando se empleó Wav2vec (14.22 %). Además, esta mejora fue considerablemente menor que con *Ladder Network*. Este resultado también debería deberse a que el modelo ya está acostumbrado a datos ruidosos debido a su preentrenamiento, por lo que el margen de mejora de los *beamformers* es menor. Además, debido a que las ponderaciones MVDR introducen un artefacto, MVDR no mejora el rendimiento del modelo [46].

Según la Tabla 4.4, la mayor degradación en CCC *Arousal*, CCC *Dominance* y CCC *Valence* para el escenario dinámico (*Recorded din*), al compararlo con el caso base de testeo (*Original*), se obtuvo con *Ladder Network*. Al igual que en el párrafo anterior, la degradación con Wav2vec es mucho menor que con *Ladder Network*. Esto corrobora la hipótesis de que, como el modelo ya ha sido entrenado con datos ruidosos, el margen de mejora de los *beamformers* es limitado. Como era de esperar, los esquemas de *beamforming* D&S y MVDR disminuyen la degradación en CCC *Arousal*, CCC *Dominance* y CCC *Valence* utilizando *Ladder Newtwork*. La tendencia en las Figuras 4.4 y 4.5 también es similar. Según la Figura 4.5, D&S y MVDR condujeron a un aumento en la suma de CCC igual a 82.97 % y 69.10 %, respectivamente, cuando se utiliza *Ladder Network*. De nuevo, sólo D&S pudo aumentar el sumatorio de CCC's cuando se empleó Wav2vec y esta mejora relativa fue muy menor que con *Ladder Network*. Esto corrobora la hipótesis antes mencionada de que debido a que las ponderaciones MVDR añaden un artefacto, MVDR no aumenta el rendimiento del modelo.

La degradación de la CCC cuando se prueba en los conjuntos de datos HRI reales frente al original es del 78.08 % y el 74.39 % para la *Ladder Network* en condiciones estáticas y dinámicas, respectivamente. Mientras tanto, para Wav2vec la degradación en las mismas condiciones es sólo del 34.66 % y el 33.66 %. La menor degradación del modelo con *transformer* podría explicarse por el hecho de que este modelo se preentrenó utilizando datos telefónicos ruidosos.

Se observa que para todas las pruebas en conjuntos de datos HRI reales (estáticos o dinámicos), al menos uno de los dos *beamformers* ayuda a mejorar la línea de base. Se observa que la única configuración en la que MVDR es mejor que D&S es para la red de escalera en la condición estática. Además, la arquitectura Wav2vec es mejor que la *ladder network* en todos los experimentos, especialmente cuando se prueba en el conjunto de datos HRI real sin utilizar ningún formador de haz.

Al comparar las tablas 4.3 y 4.6, se observa que la mejora obtenida al entrenar con el conjunto de datos simulado frente al conjunto de datos original en el escenario HRI real estático cuando se utiliza D&S, es de 27.49 % y 13.86 % para los modelos *ladder* y Wav2vec, respectivamente. Por otro lado, la mejora en la misma condición, pero usando MVDR es de 29.05 % y 18.99 % para los modelos *ladder* y Wav2vec, respectivamente.

Las tablas 4.4 y 4.7 muestran que, en la situación dinámica real del HRI, el entrenamiento con el conjunto de datos simulado en comparación con el conjunto de datos original mejoró el rendimiento en un 45.51 % y un 15.26 % para *ladder* y Wav2vec cuando se utilizó D&S, y en un 48.85 % y un 21.19 % para *ladder* y Wav2vec cuando se utilizó MVDR.

Se observa que los modelos con mayor mejora con el conjunto de datos de entrenamiento simulado son los que tenían peor rendimiento en primer lugar y, por tanto, un mayor rango de mejora.

Si se compara el mejor modelo obtenido para la condición estática con el mejor modelo obtenido para la condición dinámica (Suma CCC 1.370 vs 1.331), se observa que existe un 2.85 % de degradación.

Capítulo 5

Conclusiones

En esta tesis, se propone un nuevo sistema para afrontar la tarea de SER en escenarios de interacción humano-robot complejos. Este sistema puede aprovecharse de sensores comúnmente presentes en robots que permitan localizar la posición del usuario (como cámaras o sensores infrarojos) y direccionar de forma virtual la ganancia del arreglo de micrófonos en esa dirección mediante el uso de mecanismos de *beamforming*. Además, el sistema utiliza técnicas de modelamiento del canal acústico con el fin de crear una base de entrenamiento simulada que logra preparar al modelo para enfrentarse al escenario real de interacción humano-robot. La simulación realizada es tan fiel a la realidad que los resultados obtenidos evaluando el sistema en la partición de prueba de la base de datos simulada, son similares a los obtenidos de la base real de HRI. Esta es la primera vez que se prueba un sistema de SER en un ambiente real de Recorded dinámico.

Se comprueba el problema de la degradación del rendimiento del sistema en términos de CCC para SER continua en una situación real de HRI, en la que el robot puede estar moviendo o rotando el micrófono. Se demostró que la incorporación del modelado del canal acústico combinado con el ruido ambiental en el proceso de entrenamiento del modelo SER, puede ayudar a mejorar los resultados en términos de CCC. Este trabajo se centra en la representación del entorno acústico y la aplicación de diferentes métodos de *beamforming* para adaptar los modelos robustos existentes a escenarios reales.

El mejor resultado para la base de datos estática real se obtuvo utilizando el modelo Wav2vec entrenado con el conjunto de datos simulado y aplicando Bost, obteniendo 0.541, 0.466 y 0.363 para *dominance*, *arousal* y *valence*, respectivamente. Esto equivale a una mejora del 30.05 % respecto a la línea base del mismo modelo. Por otro lado, el mejor resultado obtenido para la base de datos dinámica se obtuvo también utilizando el modelo Wav2vec entrenado con la base de datos simulada y aplicando Bost, obteniendo 0.540, 0.459 y 0.332. Esto supone una mejora del 24 % con el caso base.

Finalmente, se concluye que el objetivo de este trabajo fue cumplido, puesto que se logró adaptar un sistema de SER tanto a ambientes HRI estáticos como dinámicos, mejorando su rendimiento de forma sustancial.

5.1. Trabajo a futuro

En primer lugar, se sugiere como trabajo a futuro la realización de los mismos experimentos, pero con el uso de métodos de mejoramiento de audio que utilicen también *deep learning*. Incluso, se puede probar realizar un entrenamiento de una etapa, es decir, entrenar al mismo tiempo el método de mejoramiento de audio con el modelo de SER. Este trabajo aún no ha sido estudiado en el área de SER, sin embargo, ha mostrado buenos resultados en tareas que también utilizan la voz como entrada principal como *speech to text*.

En segundo lugar, se propone realizar experimentos que también utilicen métodos de filtrado espacial, pero con la existencia de más de una fuente de voz. Esto supondría un mayor reto, puesto que la separación de *speakers* tiene mayor dificultad que solamente extraer ruido de un audio con un solo hablante. Este experimento estaría aún más cerca de una situación real de HRI que los experimentos planteados en esta tesis.

Bibliografía

[1] Yang, G. Z., Bellingham, J., Dupont, P. E., Fischer, P., Floridi, L., Full, R., Jacobstein, N., Kumar, V., McNutt, M., Merrifield, R., Nelson, B. J., Scassellati, B., Taddeo, M., Taylor, R., Veloso, M., Wang, Z. L., y Wood, R., "The grand challenges of science robotics," Science Robotics, vol. 3, 2018, doi:10.1126/scirobotics.aar7650.

[2] Rossi, S., Ferland, F., y Tapus, A., "User profiling and behavioral adaptation for hri: A survey," Pattern Recognition Letters, vol. 99, 2017, doi:10.1016/j.patrec.2017.06.002.

[3] Gaschler, A., Jentzsch, S., Giuliani, M., Huth, K., de Ruiter, J., y Knoll, A., "Social behavior recognition using body posture and head pose for human-robot interaction," en 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2128–2133, 2012, doi:10.1109/IROS.2012.6385460.

[4] Faria, D. R., Vieira, M., Faria, F. C., y Premebida, C., "Affective facial expressions recognition for human-robot interaction," en RO-MAN 2017 - 26th IEEE International Symposium on Robot and Human Interactive Communication, vol. 2017-January, 2017, doi:10.1109/ROMAN.2017.8172395.

[5] Deng, J., Pang, G., Zhang, Z., Pang, Z., Yang, H., y Yang, G., "Cgan based facial expression recognition for human-robot interaction," IEEE Access, vol. 7, 2019, doi:10.1109/ACCESS.2019.2891668.

[6] Paletta, L., Pszeida, M., Ganster, H., Fuhrmann, F., Weiss, W., Ladstätter, S., Dini, A., Murg, S., Mayer, H., Brijacak, I., y Reiterer, B., "Gaze-based human factors measurements for the evaluation of intuitive human-robot collaboration in real-time," en 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1528–1531, 2019, doi:10.1109/ETFA.2019.8869270.

[7] Chakraborty, P., Ahmed, S., Yousuf, M. A., Azad, A., Alyami, S. A., y Moni, M. A., "A human-robot interaction system calculating visual focus of human's attention level," IEEE Access, vol. 9, 2021, doi:10.1109/ACCESS.2021.3091642.

[8] Stock-Homburg, R., "Survey of emotions in human–robot interactions: Perspectives from robotic psychology on 20 years of research," International Journal of Social Robotics, vol. 14, pp. 389–411, 2022, doi:10.1007/s12369-021-00778-6.

[9] "Spezialetti, matteo and placidi, giuseppe and rossi, silvia," Frontiers in Robotics and AI, vol. 7, 2020, doi:10.3389/frobt.2020.532279.

[10] Ahmed, N., Aghbari, Z. A., y Girija, S., "A systematic survey on multimodal emotion recognition using learning algorithms," Intelligent Systems with Applications, vol. 17, p. 200171, 2023, doi:https://doi.org/10.1016/j.iswa.2022.200171.

[11] Sönmez, Y. y Varol, A., "The necessity of emotion recognition from speech signals for

natural and effective human-robot interaction in society 5.0," en 2022 10th International Symposium on Digital Forensics and Security (ISDFS), pp. 1–8, 2022, doi:10.1109/ISDFS55398.2022.9800837.

[12] Fahad, M. S., Ranjan, A., Yadav, J., y Deepak, A., "A survey of speech emotion recognition in natural environment," Digital Signal Processing: A Review Journal, vol. 110, 2021, doi:10.1016/j.dsp.2020.102951.

[13] Singh, Y. B. y Goel, S., "A systematic literature review of speech emotion recognition approaches," Neurocomputing, vol. 492, pp. 245–263, 2022, doi:https://doi.org/10.1016/j.neucom.2022.04.028.

[14] Salekin, A., Chen, Z., Ahmed, M. Y., Lach, J., Metz, D., Haye, K. D. L., Bell, B., y Stankovic, J. A., "Distant emotion recognition," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, 2017, doi:10.1145/3130961.

[15] Ahmed, M. Y., Chen, Z., Fass, E., y Stankovic, J., "Real time distant speech emotion recognition in indoor environments," en Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous 2017, (New York, NY, USA), p. 215–224, Association for Computing Machinery, 2017, doi:10.1145/3144457.3144503.

[16] Chen, L., Su, W., Feng, Y., Wu, M., She, J., y Hirota, K., "Two-layer fuzzy multiple random forest for speech emotion recognition in human-robot interaction," Information Sciences, vol. 509, 2020, doi:10.1016/j.ins.2019.09.005.

[17] Gunes, H. y Pantic, M., "Automatic, dimensional and continuous emotion recognition," International Journal of Synthetic Emotions, vol. 1, 2010, doi:10.4018/jse.2010101605.

[18] Lawrence, I. y Lin, K., "A concordance correlation coefficient to evaluate reproducibility," Biometrics, pp. 255–268, 1989.

[19] Rasmus, A., Berglund, M., Honkala, M., Valpola, H., y Raiko, T., "Semi-supervised learning with ladder networks," en Advances in Neural Information Processing Systems (Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., y Garnett, R., eds.), vol. 28, Curran Associates, Inc., 2015, https://proceedings.neurips.cc/paper/2015/file/378a063b8fdb1db941e34f4bde584c7d-Paper.pdf.

[20] Huang, J., Li, Y., Tao, J., Lian, Z., Niu, M., y Yi, J., "Speech emotion recognition using semi-supervised learning with ladder networks," en 2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia), pp. 1–5, 2018, doi:10.1109/ACIIAsia.2018.8470363.

[21] Parthasarathy, S. y Busso, C., "Ladder Networks for Emotion Recognition: Using Unsupervised Auxiliary Tasks to Improve Predictions of Emotional Attributes," en Proc. Interspeech 2018, pp. 3698–3702, 2018, doi:10.21437/Interspeech.2018-1391.

[22] Parthasarathy, S. y Busso, C., "Semi-supervised speech emotion recognition with ladder networks," IEEE/ACM Transactions on Audio Speech and Language Processing, vol. 28, 2020, doi:10.1109/TASLP.2020.3023632.

[23] Tao, J. H., Huang, J., Li, Y., Lian, Z., y Niu, M. Y., "Semi-supervised ladder networks for speech emotion recognition," International Journal of Automation and Computing, vol. 16, 2019, doi:10.1007/s11633-019-1175-x.

[24] Leem, S.-G., Fulford, D., Onnela, J.-P., Gard, D., y Busso, C., "Separation of Emotional

and Reconstruction Embeddings on Ladder Network to Improve Speech Emotion Recognition Robustness in Noisy Conditions," en Proc. Interspeech 2021, pp. 2871–2875, 2021, doi:10.21437/Interspeech.2021-1438.

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., y Polosukhin, I., "Attention is all you need," en Advances in Neural Information Processing Systems (Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., y Garnett, R., eds.), vol. 30, Curran Associates, Inc., 2017, https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[26] Baevski, A., Zhou, H., Mohamed, A., y Auli, M., "wav2vec 2.0: A framework for self-supervised learning of speech representations," Advances in Neural Information Processing Systems, vol. 2020-December, 2020, doi:10.48550/arxiv.2006.11477.

[27] Mohamed, A., Okhonko, D., y Zettlemoyer, L., "Transformers with convolutional context for asr," arXiv preprint arXiv:1904.11660, 2019, doi:10.48550/ARXIV.1904.11660.

[28] Berg, J., Lottermoser, A., Richter, C., y Reinhart, G., "Human-robot-interaction for mobile industrial robot teams," Procedia CIRP, vol. 79, 2019, doi:10.1016/j.procir.2019.02.080.

[29] Kousi, N., Stoubos, C., Gkournelos, C., Michalos, G., y Makris, S., "Enabling human robot interaction in flexible robotic assembly lines: An augmented reality based software suite," Procedia CIRP, vol. 81, 2019, doi:10.1016/j.procir.2019.04.328.

[30] Miseikis, J., Caroni, P., Duchamp, P., Gasser, A., Marko, R., Miseikiene, N., Zwilling, F., Castelbajac, C. D., Eicher, L., Fruh, M., y Fruh, H., "Lio-a personal robot assistant for human-robot interaction and care applications," IEEE Robotics and Automation Letters, vol. 5, 2020, doi:10.1109/LRA.2020.3007462.

[31] Novoa, J., Mahu, R., Wuth, J., Escudero, J. P., Fredes, J., y Yoma, N. B., "Automatic speech recognition for indoor hri scenarios," ACM Transactions on Human-Robot Interaction, vol. 10, 2021, doi:10.1145/3442629.

[32] Simmer, K. U., Bitzer, J., y Marro, C., Post-Filtering Techniques, pp. 39–60. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, doi:10.1007/978-3-662-04619-7_3.

[33] Díaz, A., Mahu, R., Novoa, J., Wuth, J., Datta, J., y Yoma, N. B., "Assessing the effect of visual servoing on the performance of linear microphone arrays in moving human-robot interaction scenarios," Computer Speech and Language, vol. 65, 2021, doi:10.1016/j.csl.2020.101136.

[34] Omologo, M., Matassoni, M., y Svaizer, P., Speech Recognition with Microphone Arrays, pp. 331–353. Springer Berlin Heidelberg, 2001, doi:10.1007/978-3-662-04619-7_15.

[35] Bitzer, J. y Simmer, K. U., Superdirective Microphone Arrays, pp. 19–38. Springer Berlin Heidelberg, 2001, doi:10.1007/978-3-662-04619-7_2.

[36] Farina, A., "Simultaneous measurement of impulse response and distortion with a swept-sine technique," Proc. AES 108th conv, Paris, France, 2000.

[37] Lotfian, R. y Busso, C., "Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings," IEEE Transactions on Affective Computing, vol. 10, 2019, doi:10.1109/TAFFC.2017.2736999.

[38] Busso, C., Bulut, M., Lee, C. C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee,

S., y Narayanan, S. S., "Iemocap: Interactive emotional dyadic motion capture database," Language Resources and Evaluation, vol. 42, 2008, doi:10.1007/s10579-008-9076-6.

[39] Metallinou, A., Yang, Z., chun Lee, C., Busso, C., Carnicke, S., y Narayanan, S., "The usc creativeit database of multimodal dyadic interactions: from speech and full body motion capture to continuous emotional annotations," Language Resources and Evaluation, vol. 50, 2016, doi:10.1007/s10579-015-9300-0.

[40] Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., y Verma, R., "Crema-d: Crowd-sourced emotional multimodal actors dataset," IEEE Transactions on Affective Computing, vol. 5, 2014, doi:10.1109/TAFFC.2014.2336244.

[41] Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Mortillaro, M., Salamin, H., Polychroniou, A., Valente, F., y Kim, S., "The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism," en Proc. Interspeech 2013, pp. 148–152, 2013, doi:10.21437/Interspeech.2013-56.

[42] Wagner, J., Triantafyllopoulos, A., Wierstorf, H., Schmitt, M., Eyben, F., y Schuller, B. W., "Dawn of the transformer era in speech emotion recognition: closing the valence gap," arXiv preprint arXiv:2203.07378, 2022.

[43] Hsu, W.-N., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., y Auli, M., "Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training," en Proc. Interspeech 2021, pp. 721–725, 2021, doi:10.21437/Interspeech.2021-236.

[44] Wang, Y., Boumadane, A., y Heba, A., "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," CoRR, vol. abs/2111.02735, 2021, doi:10.48550/arXiv.2111.02735.

[45] Hsu, W.-N., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., y Auli, M., "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," 2021, doi:10.48550/ARXIV.2104.01027.

[46] Erdogan, H., Hershey, J., Watanabe, S., Mandel, M., y Roux, J. L., "Improved mvdr beamforming using single-channel mask prediction networks," Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 08-12-September-2016, pp. 1981–1985, 2016, doi:10.21437/INTERSPEECH.2016-552.

Anexo

Anexo A: Lista de acrónimos

A continuación, se listan en orden alfabético los acrónimos utilizados en este documento:

ASR	Automatic Speech Recognition o Reconocimiento Automático de Voz
AOI	Angle of Incidence o Ángulo de Incidencia
CCC	Concordance Correlation Coefficient
CDR	Coherent-to-Diffuse Power Ratio Estimation
CNN	Convolutional Neural Networks o Redes Neuronales Convolucionales
D&S	Delay and Dum o Desfase y Suma
DOA	Direction of Arrival o Dirección de Llegada
FC	Fully Connected o Totalmente Conectado
GELU	Gaussian Error Linear Unit
HCI	Human-Computer Interaction o Interacción Humano-Computador
HLD	High Level Descriptors o Descriptores de Alto Nivel
HRI	Human-Robot Interaction o Interacción Humano-Robot
IDFT	Inverse Discrete Fourier Transform o Transformada Inversa de Fourier Discreta
LLD	Low Level Descriptors o Descriptores de Bajo Nivel
LPTV	Laboratorio de Procesamiento y Transmisión de Voz
LSTM	Long Short Term Memory
MLP	Multilayer Perceptron o Perceptrón Multicapa
MVDR	Minimum Variance Distortionless Response
PLN	Procesamiento del Lenguaje Natura
RIR	Room Impulse Response
ROS	Robot Operating System
SER	Speech Emotion Recognition o Reconocimiento de Emociones por Voz
SNR	Signal-to-Noise Ratio o Relación Señal-Ruido
VAD	Voice Activity Detector o Detector de Actividad de Voz